

Badanie wydajności złączeń i zapytań zagnieżdżonych na schematach znormalizowanych i zdenormalizowanych

Bujak Konrad

Kraków, 04.06.2021

1. Wstęp

Celem przeprowadzonego badania było porównanie wydajności złączeń oraz zapytań zagnieżdżonych na utworzonych tabelach o dużej liczbie danych. Wykorzystane zostały schematy relacyjnych baz danych umieszczone w artykule Łukasza Jajeńnicy i Adama Piórkowskiego pt. *WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH*.

2. Konfiguracja sprzętowa i programowa

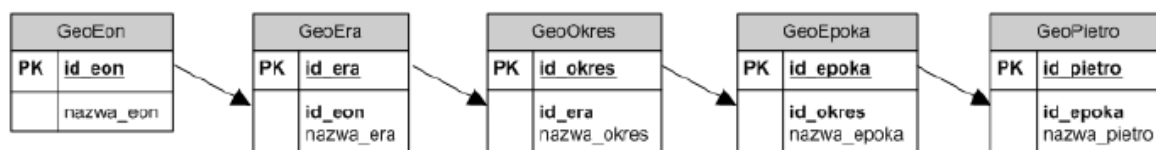
Wszystkie przeprowadzone testy zostały wykonane na podanej konfiguracji sprzętowej:

- CPU: Intel Core i5 9300H 8MB cache @ 2,4GHz up to 4,1GHz 4 cores 8 threads,
- RAM: 8GB Samsung 2666MHz,
- SSD: Kingston M.2 256 GB PCI-Express x4 NVMe,
- S.O.: Windows 10 Home 64-bit v. 20H2 19042.985
- System zarządzania bazami danych: SQL Server Management Studio v.18.8

3. Metodyka

Pierwszym krokiem było utworzenie modelu tabeli geochronologicznej na podstawie schematu znormalizowanego zamieszczonego poniżej (Rys. 1.), po czym za pomocą odpowiednio przygotowanego złączenia utworzono nową zdenormalizowaną tabelę o nazwie *GeoTabela*:

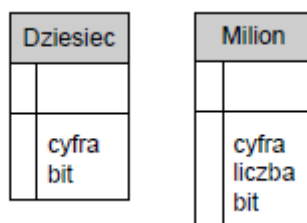
```
SELECT nazwa_eon, GeoEon.id_eon, nazwa_era, GeoEra.id_era, nazwa_okres,
GeoOkres.id_okres, nazwa_epoka, GeoEpoka.id_epoka, nazwa_pietro,
GeoPietro.id_pietro INTO GeoTabela FROM GeoPietro JOIN GeoEpoka ON
GeoPietro.id_epoka = GeoEpoka.id_epoka
JOIN GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres JOIN GeoEra ON
GeoOkres.id_era = GeoEra.id_era JOIN GeoEon ON GeoEra.id_eon = GeoEon.id_eon;
```



Rys. 1. Znormalizowany schemat tabeli geochronologicznej.

Następnie utworzono tabele *Dziesięć* oraz *Milion* na podstawie niżej zamieszczonych schematów (Rys. 2.). W tabeli *Dziesięć* pole *liczba* zostało wypełnione liczbami od 0 do 9, a pole *bit* losowo wartościami 0 i 1. Tabela *Milion* utworzona została za pomocą odpowiednio przygotowanego złączenia:

```
CREATE TABLE Milion(liczba int, cyfra int, bit int);
INSERT INTO Milion SELECT a1.cyfra +10* a2.cyfra +100*a3.cyfra + 1000*a4.cyfra
+ 10000*a5.cyfra + 10000*a6.cyfra AS liczba , a1.cyfra AS cyfra, a1.bit AS bit
FROM Dziesięć a1, Dziesięć a2, Dziesięć a3, Dziesięć a4, Dziesięć a5, Dziesięć a6 ;
```



Rys. 2. Schematy tabel *Dziesięć* i *Milion*.

W teście wykonano po cztery zapytania zawarte w przytaczanym artykule dla danych przed i po indeksacji kolumn w tabelach. Poniżej przedstawiono treść zapytania:

- Zapytanie 1 (1 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej:

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON
Milion.liczba%68=(GeoTabela.id_pietro);
```

- Zapytanie 2 (2 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej:

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON (Milion.liczba%68 =
GeoPietro.id_pietro) JOIN GeoEpoka ON GeoPietro.id_epoka = GeoEpoka.id_epoka
JOIN GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres JOIN GeoEra ON
GeoOkres.id_era = GeoEra.id_era JOIN GeoEon ON GeoEra.id_eon = GeoEon.id_eon;
```

- Zapytanie 3 (3 ZG), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane:

```
SELECT COUNT(*) FROM Milion WHERE Milion.liczba%68=
(SELECT id_pietro FROM GeoTabela WHERE Milion.liczba%68=(id_pietro));
```

- Zapytanie 4 (4 ZG), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych:

```
SELECT COUNT(*) FROM Milion WHERE Milion.liczba%68=
(SELECT GeoPietro.id_pietro FROM GeoPietro JOIN GeoEpoka ON GeoPietro.id_epoka =
GeoEpoka.id_epoka JOIN GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres JOIN
GeoEra ON GeoOkres.id_era = GeoEra.id_era JOIN GeoEon ON GeoEra.id_eon =
GeoEon.id_eon WHERE Milion.liczba%68 =id_pietro);
```

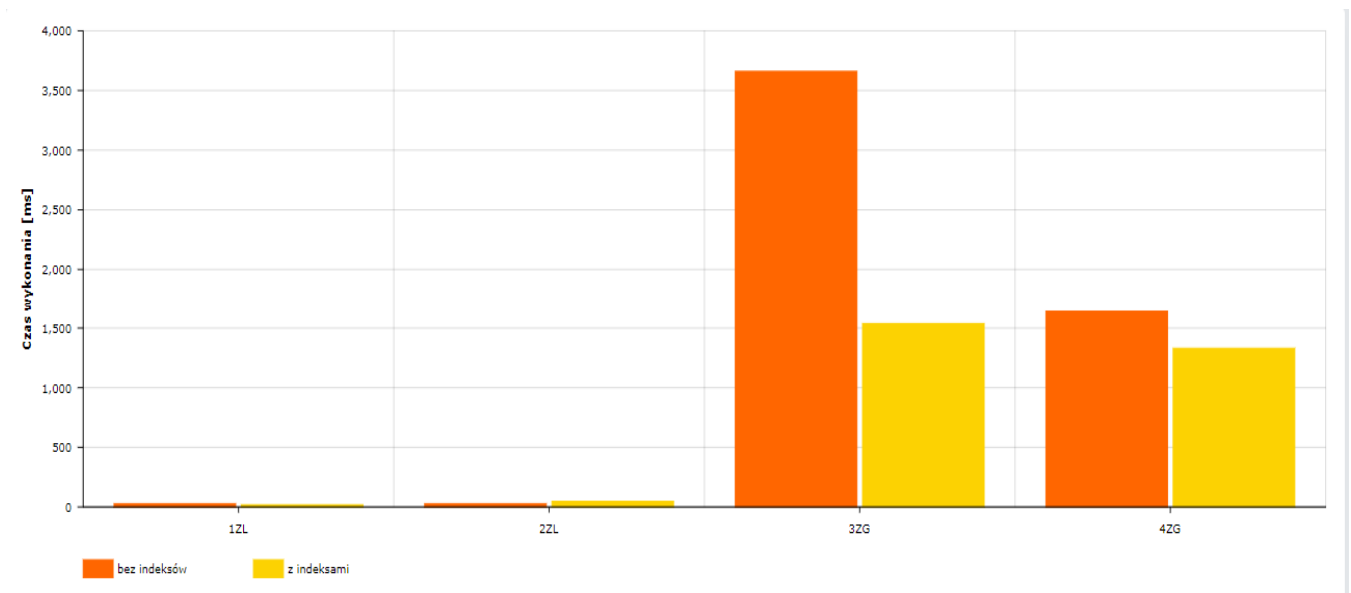
4. Wyniki

Testy wielokrotnie powtarzano, po czym odrzucano kolejne odstające wartości, aż do momentu pozostania dziesięciu wyników dla każdego zapytania. W Tabeli 1 znajdującej się poniżej przedstawiono średnią, minimalną oraz maksymalną wartość z pozostałych dziesięciu pomiarów.

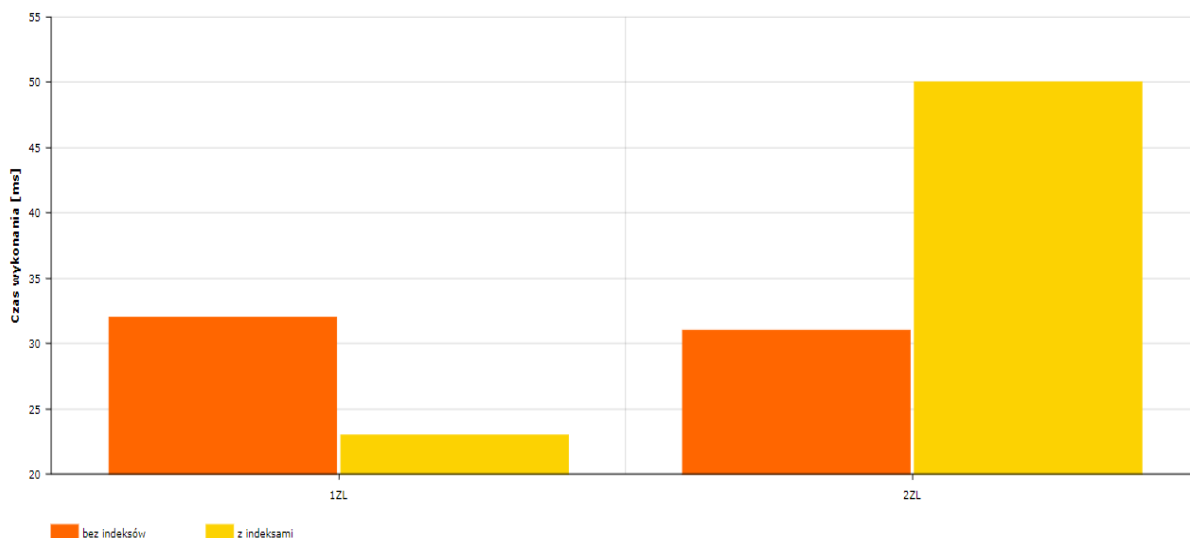
	1ZL			2ZL			3ZG			4ZG		
	min	średnia	max	min	średnia	max	min	średnia	max	min	średnia	max
bez indeksów	30	32	35	29	31	33	3470	3661	3918	1564	1643	1711
z indeksami	21	23	26	48	50	54	1446	1542	1669	1282	1336	1390

Tabela 1

Aby ułatwić analizę wyników przedstawiono je także na wykresie ogólnym (Rys. 3.) oraz wykresie dla pierwszego i drugiego zapytania (Rys. 4.).



Rys. 3.



Rys. 4.

5. Wnioski

Po analizie otrzymanych wyników otrzymano następujące wnioski:

- Złączenia są w każdym przypadku szybsze niż zapytania zagnieżdżone.
- Indeksowanie kolumn znacząco skraca czas potrzebny na wykonanie zapytań zagnieżdżonych zarówno w postaci zdenormalizowanej i znormalizowanej.
- Indeksacja kolumn w postaci znormalizowanej zdecydowanie spowolniła wykonanie złączenia.
- Postać znormalizowana jest wydajniejsza niż zdenormalizowana w każdym przypadku oprócz złączenia, w którym kolumny zostały poddane indeksacji.

Ostatecznie możemy stwierdzić, że używając podanej konfiguracji sprzętowej, w systemie zarządzania bazami danych SQL Server Management Studio v.18.8, korzystanie z postaci znormalizowanej w większości przypadków prowadzi do poprawienia wydajności. Nie sprawdza się to jedynie w wypadku indeksacji kolumn w złączeniu, gdzie czas potrzebny na wykonanie zwiększył się prawie dwukrotnie.

6. Bibliografia

1. Łukasz JAJEŚNICA, A. P. (2010, vol. 31 Number 2A (89)). WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŹDŹEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH. *STUDIA INFORMATICA*, strony 445-456.
2. Materiały używane na zajęciach, autorstwa dr inż. Michała Lupy i dr hab. inż. Adama Piórkowskiego, profesora AGH.