Sprawozdanie z obliczeń dokonanych na podstawie artkułu pt. „Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych”, Studia Informatica

Jan Skwarczeński

1. Wprowadzenie

Celem sprawozdania jest przedstawienie wyników testów wydajności zapytań dla aspektów normalizacji oraz indeksowania baz danych. Rezultaty eksperymentów oparto o porównanie dwóch systemów zarządzania bazami danych - Microsoft SQL Server oraz PostgreSQL. Badania były przeprowadzone na przykładzie tabeli geochronologicznej oraz tablicach pomocniczych zapełnionych liczbami.

2. Konfiguracja sprzętowa

Jednostka na której zostały przeprowadzone testy:

- CPU: Intel Core i5-8265U (4 rdzenie, 8 wątków, 1.60-3.90 GHz, 6 MB cache)
- GPU: Intel UHD Graphics 620
- RAM: 8 GB (DDR4, 2400MHz)
- SSD M.2: LiteOn CL1-3D256-Q11 (2100 MB/s / 800 MB/s)
- System operacyjny: Windows 10 Pro 21H2 64-bit

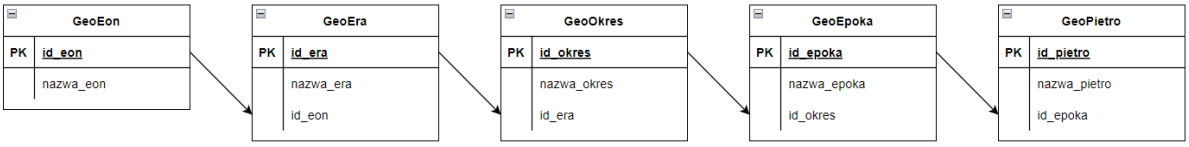
Wszelkie testy zostały wykonane przy ustawieniu komputera w najwydajniejszym trybie.

Systemy zarządzania bazami danych:

- SQL Server Management Studio v18.11 (15.0.18404.0)
- PostgreSQL 14.3 build 1914

3. Przygotowanie danych

W pierwszym etapie została utworzona tabela geochronologiczna w postaci znormalizowanej (postać płatka śniegu). Obliczenia oparto na modelu tabeli ustalony przez Międzynarodową Komisję Stratygrafii (ISC) w wersji 2022/2. Podczas obliczeń został wzięty pod uwagę jedynie okres czasu odpowiadający fanerozoiku.

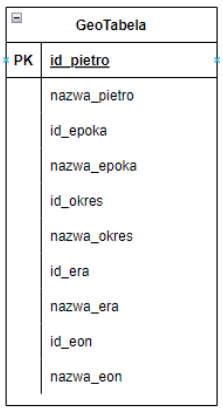


Rysunek 1: Schemat tabeli geochronologicznej

Znormalizowany schemat tabeli geochronologicznej jest podzielony na 5 tabeli:

- **GeoEon** – 1 element,
- **GeoEra** – 3 elementy,
- **GeoOkres** – 12 elementów,
- **GeoEpoka** – 34 elementy,
- **GeoPietro** – 102 elementy.

Kolejnym krokiem było utworzenie tabeli **GeoTabela** w formie zdemoralizowanej (schemat gwiazdy). Dokonano tego na podstawie złączania INNER JOIN (w przypadku MS SQL Server) oraz NATURAL JOIN (w przypadku PostgreSQL), obejmując wszystkie tabele tworzące hierarchię.



Rysunek 2: GeoTabela

GeoTabela podobnie jak GeoPietro składa się z 102 elementów.

Następnie stworzono tabele **Dziesiec** oraz **Milion**, aby móc dokładniej sprawdzić wydajność złączeń oraz zapytań zagnieżdżonych. Tabela Dziesiec była wypełniona cyframi 0 – 9 w postaci dziesiętnej oraz binarnej i służyła do utworzenia tabeli Milion która zawierała milion elementów w postaci liczb od 0 do 999 999.

4. Testy

Testy wydajnościowe były podzielone na dwie części:

- pierwsza część obejmowała zapytania bez nałożonych dodatkowych indeksów (jedynymi indeksowanymi danymi były dane w kolumnach będących kluczami głównymi poszczególnych tabel),
- w drugiej części nałożono indeksy na wszystkie kolumny biorące udział w złączeniu (tj. id_eon z tabeli GeoEra, id_era z tabeli GeoOkres, id_okres z tabeli GeoEpoka oraz id_epoka z tabeli GeoPietro) i ponownie wykonano zapytania.

Zapytania:

- **Zapytanie 1** – złączenie syntetyczne tablicy Milion z GeoTabela dodając do warunku złączenia operację modulo, dopasowującą zakresy wartości złączanych kolumn,
- **Zapytanie 2** – złączenie syntetyczne tablicy Milion z tabelą geochronologiczną w postaci znormalizowanej,
- **Zapytanie 3** – złączenie syntetyczne tablicy Milion z GeoTabela, ale złączenie jest wykonywane przez zagnieżdżenie skorelowane,
- **Zapytanie 4** – złączenie syntetyczne tablicy Milion z tabelą geochronologiczną w postaci znormalizowanej, ale złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych.

5. Wyniki testów

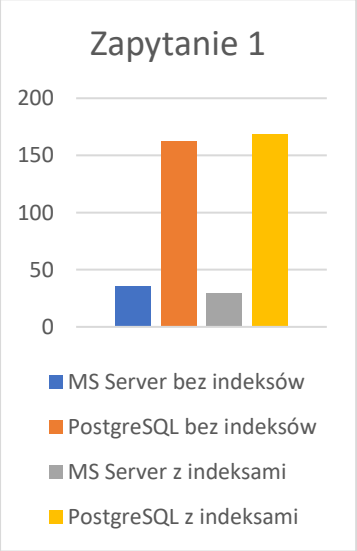
Każdy z testów został przeprowadzony dziesięciokrotnie. Zostały pomięte wyniki mogące być błędem pomiarowym, wynikającym z chwilowego spadku mocy obliczeniowej poświęconej na wykonanie zapytań na koszt wykonania inny procesów w komputerze. Zostały obliczone wartości średnie oraz minimalne czasy wykonanie

każdego zapytania. W przypadku obu systemów zarządzania bazami danych stabilności i powtarzalność testów była na podobnym poziomie.

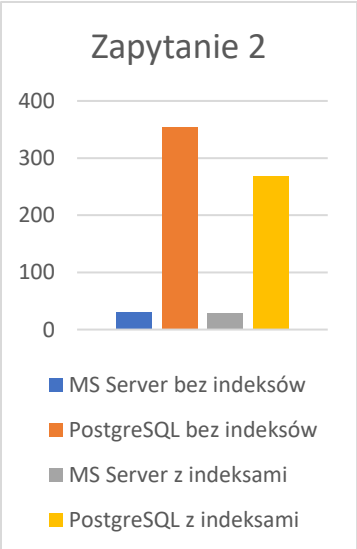
	1 ZL		2 ZL		3 ZL		4 ZL	
BEZ INDEKSÓW	**MIN**	**SR**	**MIN**	**SR**	**MIN**	**SR**	**MIN**	**SR**
SQL SERVER	29	35.8	27	30.8	27	32.6	28	34.9
POSTGRESQL	152	162.6	304	354.6	11426	12096.6	151	170.4
Z INDEKSAMI	**MIN**	**SR**	**MIN**	**SR**	**MIN**	**SR**	**MIN**	**SR**
SQL SERVER	27	29.6	26	29.2	29	30.6	28	30.4
POSTGRESQL	136	168.8	240	267.8	11356	12004.4	149	183

Tabela 1: Wyniki testów podane w milisekundach

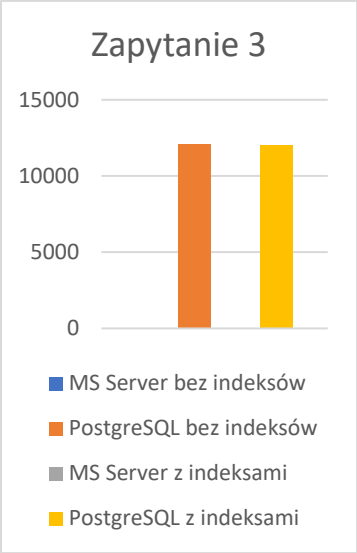
Poniższe wykresy zostały oparte na wartościach średnich.



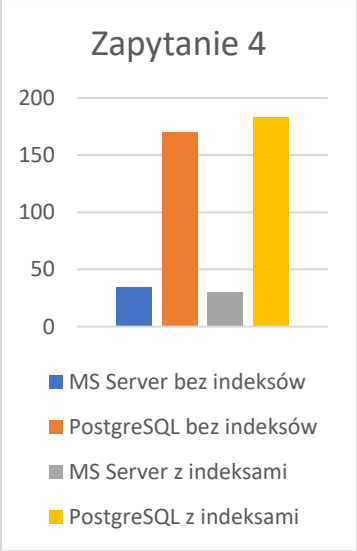
Dla zapytania pierwszego czas wykonania w MS Server jest około pięciokrotnie niższy niż w PostgreSQL. Po nałożeniu indeksów w przypadku MS Server zyskujemy 6.2 ms. W przypadku PostgreSQL ta operacja powoduję nieznaczne zwolnienie procesu.



W przypadku zapytania drugiego czasy dla MS Server są do siebie bardzo zbliżone. Różnice mogą wynikać z błędów pomiarowych. Jednak w przypadku PostgrSQL widać wyraźną różnice na korzyść tabeli z indeksami, która przy dużej ilości danych może znacząco skrócić czas pracy.



W zapytaniu 3 są widoczne największe różnice pomiędzy opracowywanymi systemami zarządzania bazami danych (około 400 krotnie dłuższy czas obliczania dla PostgreSQL). Poindeksowanie tablic daje nieznaczne różnice biorąc pod uwagę skalę.



Dla zapytania 4 w przypadku MS Server korzystniejszy czas otrzymujemy dla operacji wykonywanych na danych z indeksami (około 15%). Obliczenia w PostgreSQL zajmują znacznie dużej, jednak wykonują się szybciej na danych bez indeksów (o około 7%).

6. Wnioski

- Postać zdenormalizowana w większości przypadków jest wydajniejsza od postaci znormalizowanej. Wyjątek jednak stanowi porównanie zapytania 3 oraz 4 w systemie PostgreSQL, czyli operacji związanej z zagnieżdżeniem skorelowanym. Dla postaci znormalizowanej otrzymujemy tam kilkukrotnie dłuższy czas. Jednak należy pamiętać, że postać znormalizowana jest łatwiejsza w późniejszej edycji i rozbudowie.
- Użycie indeksów w MS Server zawsze przynosiło pozytywny skutek, jednak w dwóch przypadkach był on nieznaczny i mieścił się w granicach błędu

pomiarowego. W PostgreSQL wyraźną korzyść użycia indeksów było widać tylko podczas zapytania 2.

- System bazo danowy PostgreSQL z przygotowanymi testami radził sobie w każdym przypadku kilkukrotnie mniej wydajnie. Największa różnica była w przypadku operacji związanej z zagnieżdżeniami skorelowanymi (zapytanie 3).

BIBLIOGRAFIA

1. WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH, STUDIA INFORMATICA Volume 31 2010 Number 2A (89)
2. https://stratigraphy.org/ - tabela stratygraficzna