



Ćwiczenie 9

Adam Biedrzycki 408228

1. WSTĘP

Ćwiczenie 9 polega na sprawdzeniu wydajności dla złączeń i zagnieżdżeń skorelowanych dla systemów zarządzania bazami danych SQL Server i PostgreSQL. Ćwiczenie bazuje na artykule Pana mgr inż. Łukasza Jajeńnicy i Pana dr hab. inż. Adama Piórkowskiego. „WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH”.

Zgodnie z artykułem utworzona została baza danych „Tabela_Geochronologiczna”.

Tabela geochronologiczna obrazuje przebieg historii Ziemi na podstawie następstwa procesów i warstw skalnych.

Do stworzenia tabeli geochronologicznej wykorzystano schemat zdenormalizowany (schemat gwiazdy), poprzez utworzenie jednej tabeli o nazwie GeoTabela zawierającej dane z tabel: GeoEon, GeoEra, GeoOkres, GeoEpoka i GeoPietro. Dokonano tego za pomocą złączenia naturalnego, obejmującego wszystkie tabele tworzące hierarchię:

```
CREATE TABLE GeoTabela AS (SELECT * FROM GeoPietro NATURAL JOIN
GeoEpoka NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN
GeoEon );
```

Utworzenie tabeli GeoTabela umożliwia szybki dostęp do wszystkich danych tabeli geochronologicznej za pomocą jednego zapytania prostego, co nie jest możliwe w przypadku schematu znormalizowanego opisanego w punkcie pierwszym.

Następnie przeprowadzono test wydajności porównującej wydajność złączeń oraz zapytań zagnieżdżonych wykonanych na tabelach o dużej ilości danych. Utworzono tabele o nazwie Milion, która została wypełniona kolejno liczbami od 1 do 999 999, oraz tabela o nazwie Dziesiec, wypełniona liczbami od 0 do 9, na podstawie której powstała tabela Milion.

2. KONFIGURACJA SPRZĘTOWA

Testy wykonano na komputerze o specyfikacji:

- CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- RAM: Pamięć DDR4 32 GB
- SSD: 256GB
- S.O.: Windows 11 Home wersja 21H2

Jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

- SQL Server Management Studio, wersja 18.11.1
- PostgreSQL 14 (pgAdmin 6.7)

3. TESTY WYDAJNOŚCI

Następnie przeprowadzany był zestaw zapytań, w celu sprawdzenia wydajności złączeń i zagnieżdżeń z tabelą geochronologiczną. Na początku zapytania były przeprowadzone bez indeksów nałożonych na kolumny danych, następnie zostały nałożone indeksy na wszystkie kolumny biorące udział w złączeniu. Zastosowano cztery poniższe zapytania:

- Zapytanie 1 (1 ZL)

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

- Zapytanie 2 (2 ZL)

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON  
(mod(Milion.liczba,68)=GeoPietro.id_pietro)  
  
NATURAL JOIN GeoEpoka NATURAL JOIN GeoOkres NATURAL JOIN GeoEra  
NATURAL JOIN GeoEon;
```

- Zapytanie 3 (3 ZG)

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

- Zapytanie 4 (4 ZG)

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,68)=  
(SELECT GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka  
NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

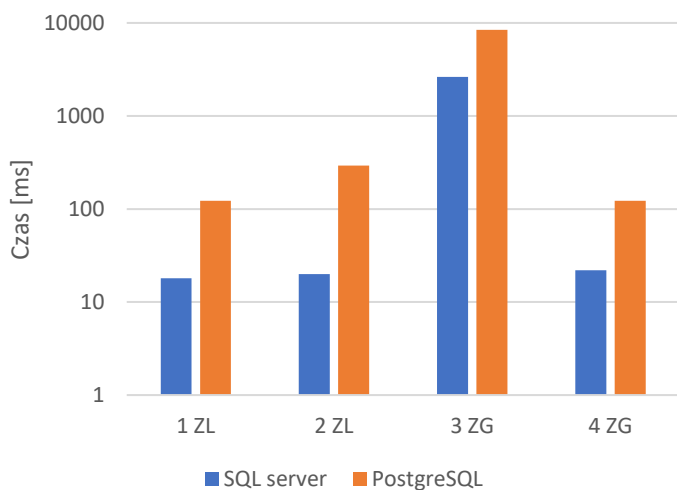
4. WYNIKI TESTÓW

Wyniki przedstawione w tabeli:

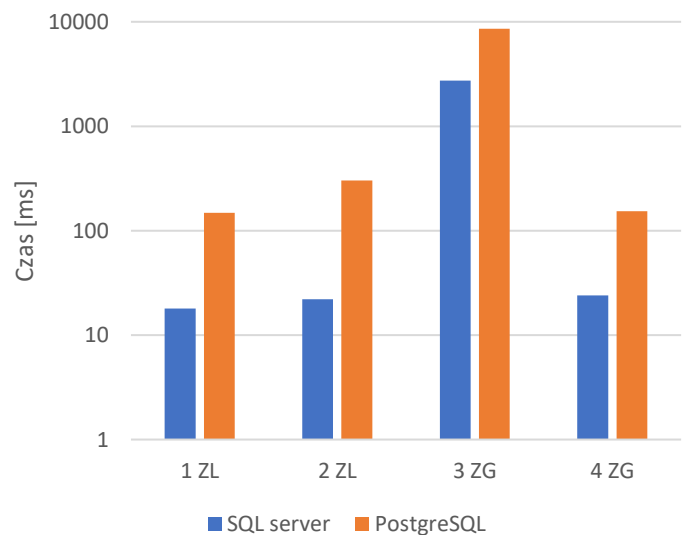
Czasy wykonania zapytań 1 ZL, 2 ZL, 3 ZG i 4 ZG [ms]

	1 ZL		2 ZL		3 ZG		4 ZG	
Bez indeksów	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
PostgreSQL	123	148	293	303,2	8438	8598	123	154
SQL server	18	18	20	22	2630	2732,2	22	24
Z indeksami	1 ZL		2 ZL		3 ZG		4 ZG	
PostgreSQL	112	129,2	201	217,4	8152	8342	116	141
SQL server	13	14	16	18,2	1245	1267	17	18

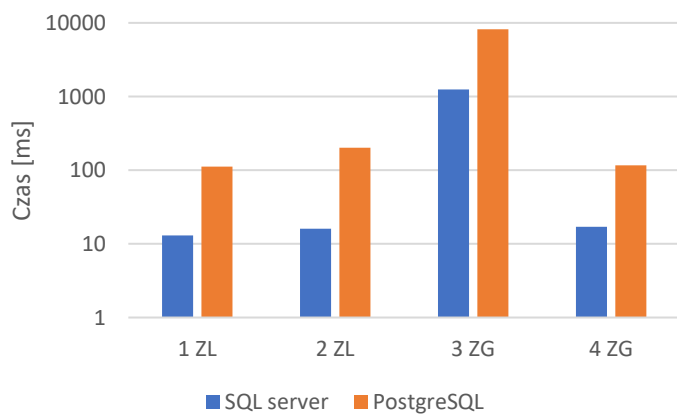
MINIMALNY CZAS WYKONANIA
ZAPYTAŃ
BEZ INDEKSÓW



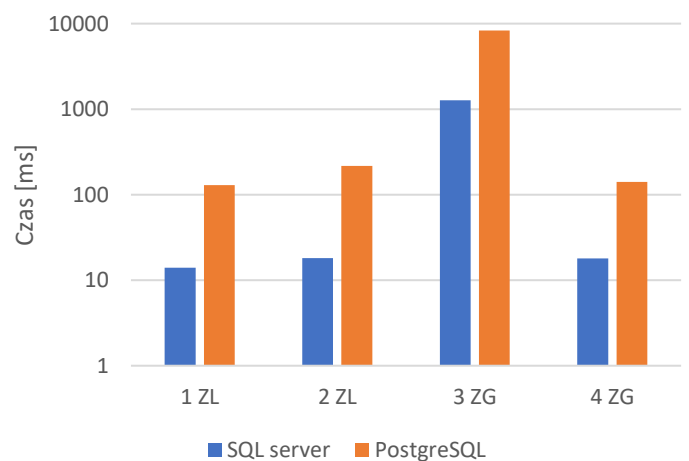
ŚREDNI CZAS WYKONANIA ZAPYTAŃ
BEZ INDEKSÓW



MINIMALNY CZAS WYKONANIA
ZAPYTAŃ
Z INDEKSAMI



ŚREDNI CZAS WYKONANIA ZAPYTAŃ
Z INDEKSAMI



5. WNIOSKI

Od razu zauważamy, że czas wykonywania zapytań w PostgreSQL jest znacznie dłuższy, niż w SQL Server. Na wykresach w osiach pionowych zastosowano wartości w skali logarytmicznej, co wizualnie łagodzi różnice między wartościami dla obu programów. Odczytując wartości z tabeli, widzimy jak duże są różnice.

Dodanie indeksów przyspieszyło czas wykonania operacji zarówno dla PostgreSQL, jak i dla SQL server. Przyspieszenie odnotowane zostało w przypadku każdego z czterech zapytań. Podsumowując, po zaznajomieniu się z charakterystyką danej bazy, warto korzystać z indeksów w celu usprawnienia wydajności pracy.