SPRAWOZDANIE

WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

Karolina Kucharz
nr albumu: 415613
Geoinformatyka II rok
Bazy Danych
Semestr letni 2023/2024

1. Wstęp

Sposób formułowania zapytań w języku SQL ma bezpośredni wpływ na wydajność bazy danych. W szczególności, semantyka zapytania odgrywa znaczącą rolę w zapewnieniu płynności oraz oszczędności zasobów podczas pracy z bazą danych. Poprawne użycie złączeń (JOIN) oraz zagnieżdżeń (subqueries) w zapytaniach SQL może znacząco wpłynąć na czas wykonania tych zapytań oraz na ogólną wydajność systemu bazodanowego. W ramach niniejszego sprawozdania skupiono się na analizie różnych rodzajów złączeń oraz zagnieżdżeń w języku SQL w celu określenia najkorzystniejszej czasowo i obliczeniowo metodyki wykonywania prac na bazach danych.

2. Cel zadania

Celem wykonanego zadania było przetestowanie wydajności kwerend bazujących na złączeniach i zagnieżdżeniach na przykładzie tabeli geochronologicznej. Analizowane zapytania odbywały się na tabelach o charakterze zarówno znormalizowanym jak i zdenormalizowanym.

3. Konfiguracja sprzętowa i programowa

Testy wykonane zostały na sprzęcie:

- Procesor CPU: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
- Pamięć RAM: 8 GB
- System operacyjny: Windows 10 wersja 22H2
- Dysk SSD: CT500BX500SSD1

Wykorzystane systemy zarządzania bazami danych:

- Microsoft SQL Server 2022 (RTM) - 16.0.1000.6 (X64)
- PostgreSQL 16.3, compiled by Visual C++ build 1939, 64-bit

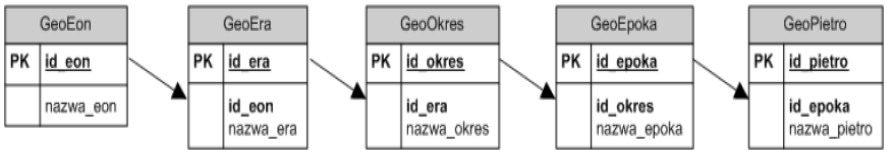
4. Wykonanie zadania

Na potrzebę realizacji analizy zbudowana została baza danych geochronologicznych na podstawie tabeli stratygraficznej zawierającej informacje na temat eonu, er, okresów i epok (Rysunek 1). Stworzonych zostało 5 tabel odpowiadających poszczególnym jednostkom czasowym, które zostały złączone ze sobą za pomocą kluczy głównych tworząc tym schemat znormalizowany (Schemat 1). Tabele te dodatkowo poddano denormalizacji do postaci jednej tabeli o nazwie GeoTabela (Schemat 2). Ponadto tabela uzupełniona została o okresy od Kambru do Syluru, epoki od Terrenewu do Przydolu oraz piętra od początku Kambru do Holocenu (w liczbie 99). Naniesiono również poprawki literówek zawartych w schemacie, na którym opiera się analiza.

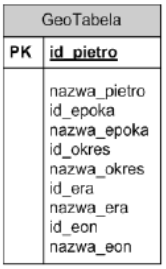
Tabela geochronologiczna

Wiek (mln lat)	Eon	Era	Okres		Epoka
0,010	FANEROZOIK	Kenzoik	Czwar-torząd		Halocen
1,8					Plejstocen
22,5			Trzeciorząd	Neo-gen	Pliocen
					Miocen
65				Paleogen	Oligocen
					Eocen
					Paleocen
140		Mezozoik	Kreda		Górna
					Dolna
195			Jura		Górna
					Środkowa
					Dolna
230			Trias		Górna
					Środkowa
					Dolna
280		Paleozoik	Perm		Górny
					Dolny
345			Karbon		Górny
					Dolny
395			Dewon		Górny
					Środkowy
					Dolny

Rysunek 1

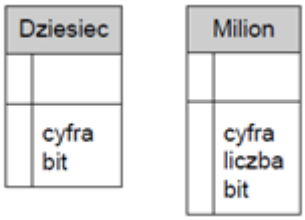


Schemat 1



Schemat 2

Do wykonania testów wydajności stworzono dwie tabele pomocnicze: pierwszą wypełnioną dziesięcioma rekordami (liczby naturalne od 0 do 9) oraz drugą wypełnioną milionem rekordów (liczby naturalne od 0 do 999 999) (Schemat 3).



Schemat 3

W trakcie realizacji zadania wykonano zapytania sprawdzające wydajność złączeń i zagnieżdżeń wykorzystujących łączenie danych tabeli geochronologicznej w wersji zdenormalizowanej i znormalizowanej z danymi z tabeli zawierającej milion rekordów. W pierwszej kolejności analizie poddano zapytania bez nałożonych indeksów na kolumny danych, gdzie jedynymi indeksowanymi danymi były dane będące kluczami głównymi dla poszczególnych tabel. W drugim etapie indeksy założone zostały na wszystkie kolumny poddane analizie. Przeprowadzone testy miały na celu określenie wpływu normalizacji danych na czas wykonywania zapytań złożonych w postaci:

- 1Z – zapytanie, którego celem było złączenie tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn:

```
/*1Z*/
SELECT COUNT(*)
FROM Milion
INNER JOIN GeoTabela ON Milion.liczba % 99 = GeoTabela.id_pietro;
```

Postać zapytania 1Z w SQL Server

- 2Z – zapytanie, którego celem było złączenie tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel:

```
/*2Z*/
SELECT COUNT(*)
FROM Milion
INNER JOIN stratii.GeoPietro
ON Milion.liczba % 99 = GeoPietro.id_pietro
INNER JOIN stratii.GeoEpoka
ON GeoPietro.id_epoka = GeoEpoka.id_epoka
INNER JOIN stratii.GeoOkres
ON GeoEpoka.id_okres = GeoOkres.id_okres
INNER JOIN stratii.GeoEra
ON GeoOkres.id_era = GeoEra.id_era
INNER JOIN stratii.GeoEon
ON GeoEra.id_eon = GeoEon.id_eon;
```

Postać zapytania 2Z w SQL Server

- 3ZG – zapytanie, którego celem było złączenie tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie było wykonywane poprzez zagnieżdżenie skorelowane:

```
/*3ZG*/
SELECT COUNT(*) FROM Milion
WHERE Milion.liczba % 99 =
(SELECT id_pietro FROM GeoTabela WHERE Milion.liczba % 99=id_pietro);
```

Postać zapytania 3ZG w SQL Server

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

```
/*4ZG*/
SELECT COUNT(*)
FROM Milion
WHERE Milion.liczba % 99 IN (
    SELECT GeoPietro.id_pietro
    FROM stratii.GeoPietro
    INNER JOIN stratii.GeoEpoka
    ON GeoPietro.id_epoka = GeoEpoka.id_epoka
    INNER JOIN stratii.GeoOkres
    ON GeoEpoka.id_okres = GeoOkres.id_okres
    INNER JOIN stratii.GeoEra
    ON GeoOkres.id_era = GeoEra.id_era
    INNER JOIN stratii.GeoEon
    ON GeoEra.id_eon = GeoEon.id_eon
);
```

Postać zapytania 4ZG w SQL Server

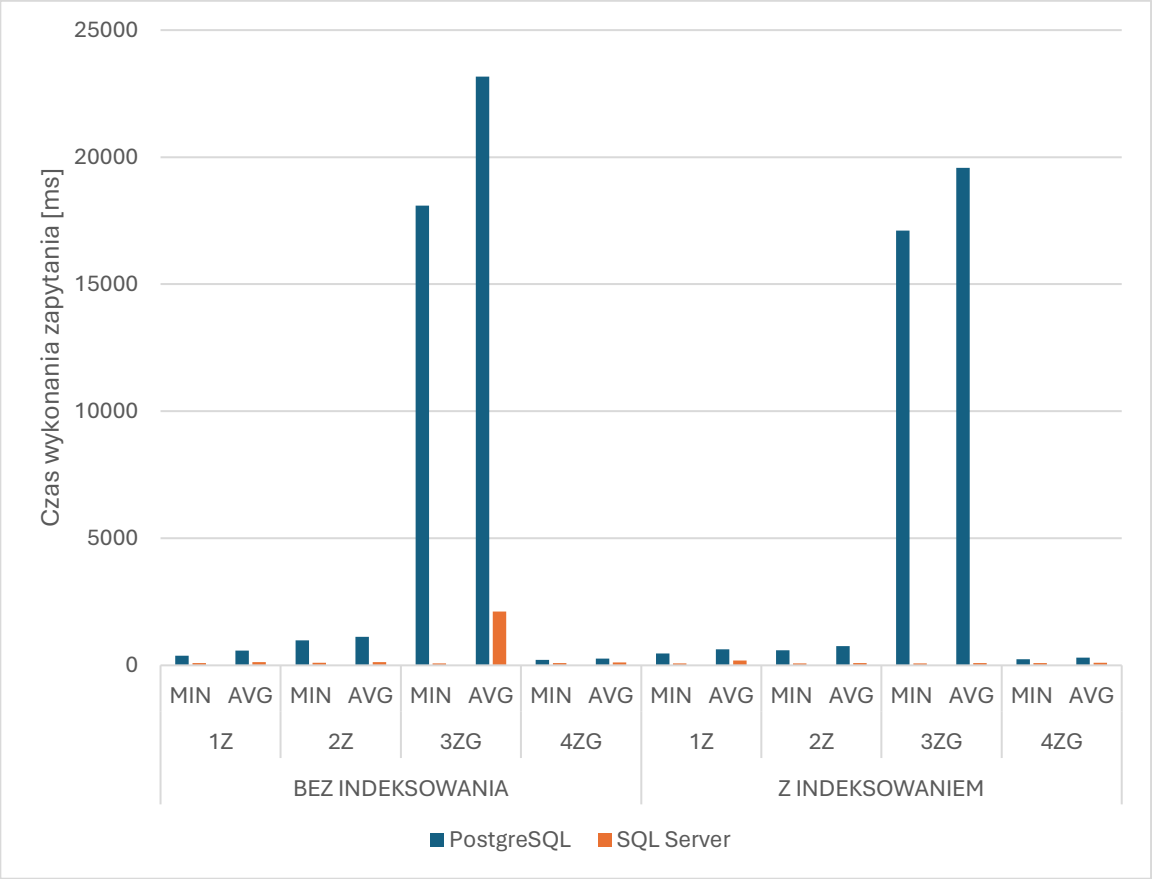
5. Wyniki

Dla każdej konfiguracji test przeprowadzono przynajmniej dziesięciokrotnie. Ostateczne wyniki zawierające minimalną pomierzoną wartość czasu realizacji zapytania oraz średni czas jego realizacji zostały zestawione w tabeli (Tabela 1). Czas wyrażony został w milisekundach.

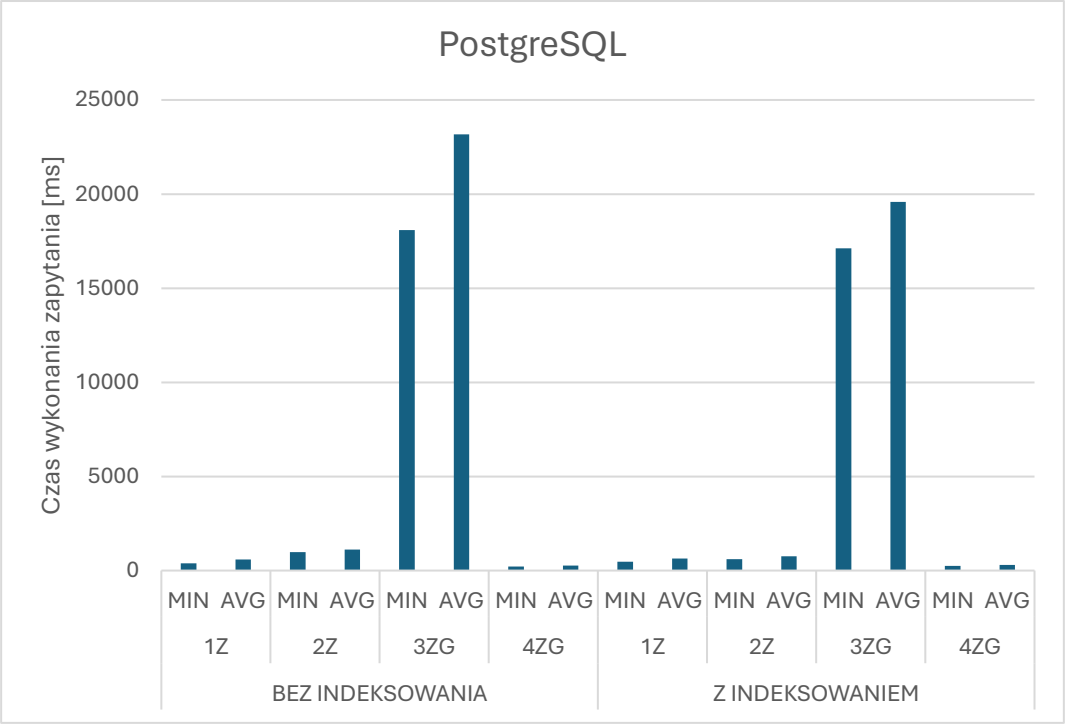
	BEZ INDEKSOWANIA								*Z INDEKSOWANIEM*							
	1Z		**2Z**		**3ZG**		**4ZG**		**1Z**		**2Z**		**3ZG**		**4ZG**	
	MIN	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*	*MIN*	*AVG*
PostgreSQL	384	581,9	979	1122,2	18091	23167,2	213	268	464	635,2	596	759,5	17113	19579,4	246	305,5
SQL Server	85	132,2	106	127,3	81	2112,4	88	112	78	190,3	82	90,2	79	92,8	89	107

Tabela 1

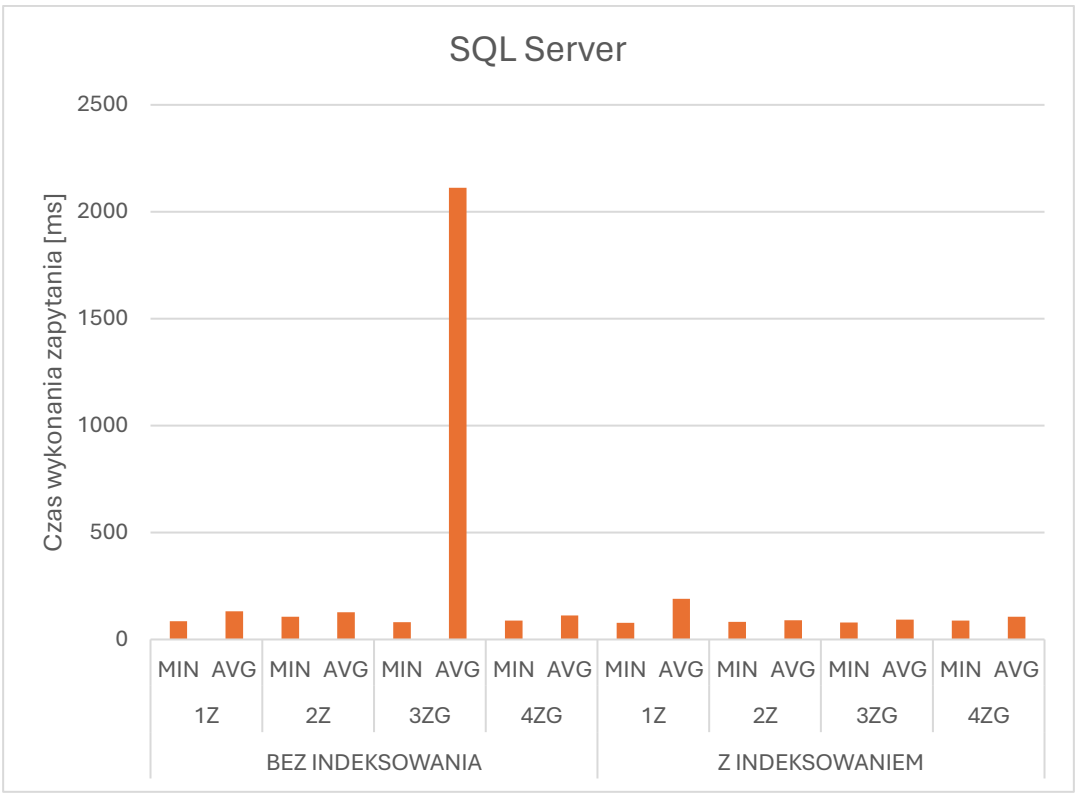
Na potrzeby analizy porównawczej dokonano również wizualizacji uzyskanych wyników w formie wykresów słupkowych.



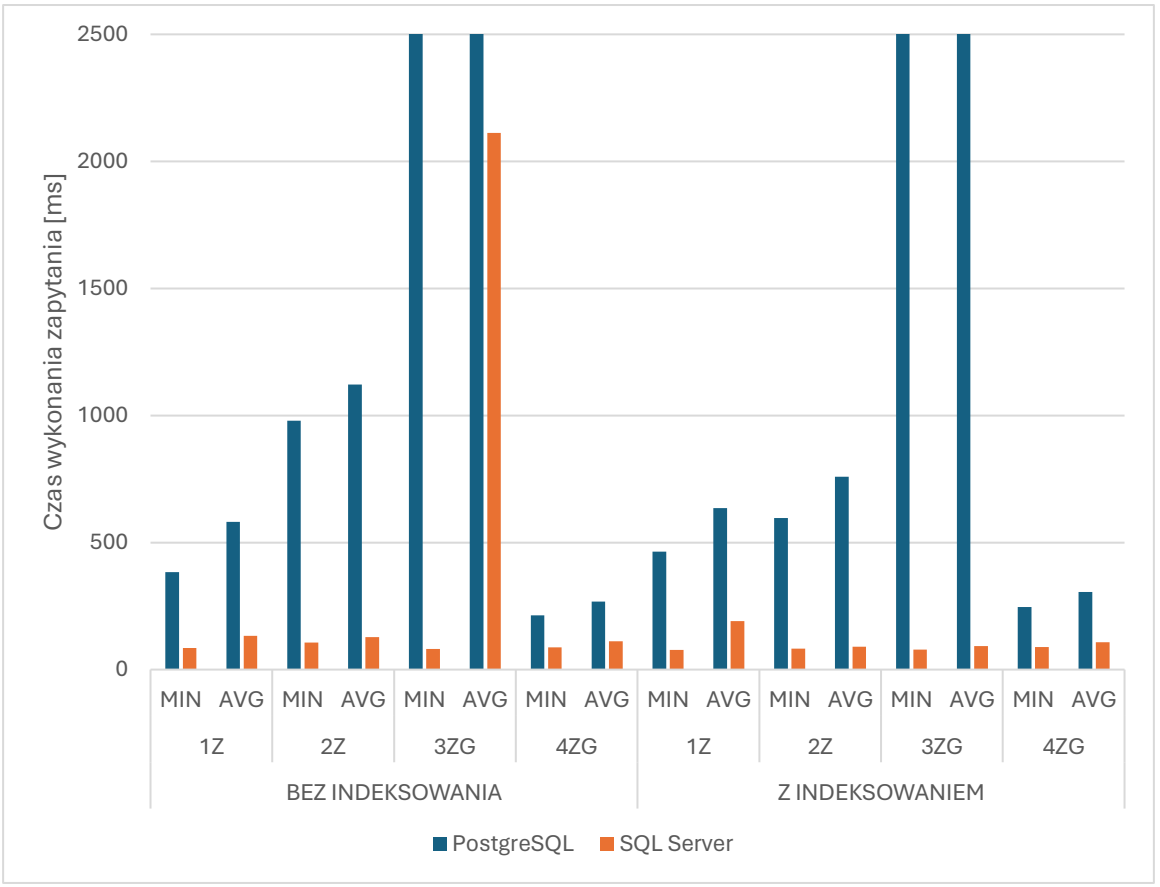
Wykres 1. Wykres zbiorczy przedstawiający wszystkie uzyskane wyniki



Wykres 2. Wykres przedstawiający dane uzyskane dla systemu PostgreSQL



Wykres 3. Wykres przedstawiający dane uzyskane dla systemu SQL Server



Wykres 4. Wykres zbiorczy przedstawiający wszystkie uzyskane wyniki ze zmodyfikowanymi wartościami osi przedstawiającej czas wykonania zapytania.

6. Wnioski

Zauważalna jest znaczna różnica w czasie wykonywania zapytań pomiędzy systemem PostgreSQL a SQL Server, widoczna w szczególności na wykresie zbiorczym (Wykres 1). Warto również zwrócić uwagę na różnicę zakresów wartości między osiami przedstawiającymi czas wykonywania zapytań dla obu systemów (Wykres 2,3).

Analizując wykresy 2 i 4 można wysnuć wniosek, iż indeksowanie pozwoliło na skrócenie czasu wykonywania zapytań w systemie PostgreSQL. Ciekawą obserwacją jest to, iż indeksowanie niemalże nie wpłynęło na wyniki uzyskane w systemie SQL Server, który dla prawie wszystkich zapytań wynik otrzymywał po około 120 ms. Jedynym wyjątkiem jest zapytanie 3ZG wykonane bez indeksowania, gdzie średni czas wykonana zapytania wyniósł ponad 2000 ms, co i tak jest wynikiem zadowalającym.

Na podstawie wyników zapytania 3ZG dla systemu PostgreSQL można stwierdzić, iż korzystanie z zagnieżdżeń skorelowanych jest metodą o niskiej wydajności czasowej niezależnie od indeksowania. Również w przypadku SQL Server zapytanie to wykonane na tablicach bez indeksowania było testem, który uzyskał najgorszy wynik spośród wszystkich wykonanych testów.

Opierając się o uzyskane rezultaty nie można jednoznacznie stwierdzić wpływu denormalizacji tabeli (zastosowanej dla zapytań 1Z oraz 3ZG) na wyniki uzyskane dla zapytań bez indeksowania. W przypadku zapytań z indeksowaniem można stwierdzić, iż brak denormalizacji pozwolił na nieznaczne skrócenie czasu wykonywania zapytań dla obu testowanych systemów.

7. Uwagi

Zadanie zrealizowano w oparciu o artykuł „Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych" Łukasza Jajeśnicy oraz Adama Piórkowskiego pod auspicjami Katedry Geoinformatyki i Informatyki Stosowanej działającej przy wydziale Geologii, Geofizyki i Ochrony Środowiska na Akademii Górniczo – Hutniczej im. Stanisława Staszica w Krakowie.