

WYDZIAŁ GEOLOGII, GEOFIZYKI I OCHRONY ŚRODOWISKA



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

*PORÓWNANIE WYDAJNOŚCI ZŁĄCZEŃ I
ZAGNIEŹDŹEŃ DLA SCHEMATÓW
ZNORMALIZOWANYCH
I ZDENORMALIZOWANYCH*

Agnieszka Ramian
Nr. albumu: 412180

Kraków, 2023

1. Wstęp

Celem projektu było zbadanie oraz porównanie wydajności zapytań bazujących na złączeniach i zagnieżdżeniach w dwóch różnych systemach zarządzania bazami danych, MySQL i PostgreSQL. Analizę przeprowadzono dla tabeli geochronologicznej.

2. Konfiguracja sprzętowa i programowa

Wszystkie testy przeprowadzono na komputerze o poniższych parametrach:

CPU: 12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz

RAM: 16GB

S.O.: Windows 11 Home.

Do zarządzanie bazami danych wykorzystane zostały oprogramowania:

PostgreSQL, wersja 15.2-2.

MySQL, wersja Community Server 8.0.33.0

3. Przygotowanie danych

W celu przeprowadzenia testów utworzono tabele odwzorowujące tabelę geochronologiczną w wersji znormalizowanej oraz zdenormalizowanej. Posłużono się także tabelą syntetycznych danych, zawierającą kolejne liczby naturalne od 0 do 999 999.

4. Wykonanie testów

Testy wydajności przeprowadzono poprzez wykonanie czterech zapytań kolejno, bez nałożonych indeksów na kolumny danych oraz z nałożonymi indeksami. Każdy z testów przeprowadzono wiele razy w dwóch oprogramowaniach.

- Test 1 (1ZL)

Celem pierwszego testu jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości łączanych kolumn przy użyciu zapytania:

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

- Test 2 (2ZL)

Celem drugiego testu jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel przy użyciu zapytania:

```
SELECT COUNT(*) FROM geo.Milion
INNER JOIN geo.Pietro ON (mod(geo.Milion.liczba,68)=geo.Pietro.id_pietro)
NATURAL JOIN geo.Epoka NATURAL JOIN geo.Okres
NATURAL JOIN geo.Era NATURAL JOIN geo.Eon;
```

- Test 3 (3ZG)

Celem trzeciego zapytania jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane przy użyciu zapytania:

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

- Test 4 (4ZG)

Celem czwartego testu 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 przy użyciu zapytania:

```
SELECT COUNT(*) FROM geo.Milion WHERE mod(geo.Milion.liczba,68)IN
(SELECT geo.Pietro.id_pietro FROM geo.Pietro
NATURAL JOIN geo.Epoka NATURAL JOIN geo.Okres
NATURAL JOIN geo.Era NATURAL JOIN geo.Eon);
```

W celu przeprowadzenia analizy mierzono w milisekundach czas wykonywania każdego z przeprowadzonych zapytań.

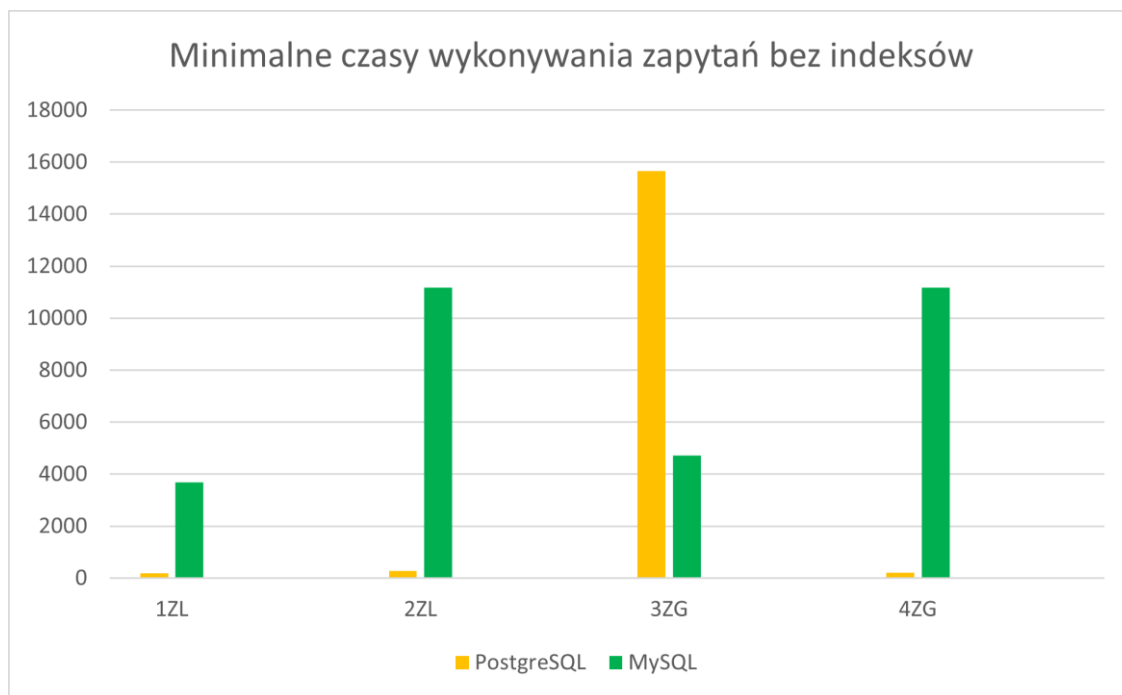
5. Wyniki

Każdy test przeprowadzono dziesięć razy. Otrzymane wartości uśredniono i w poniższej tabeli zestawiono z najkrótszymi zmierzonymi czasami wykonywania zapytań (tabela 1).

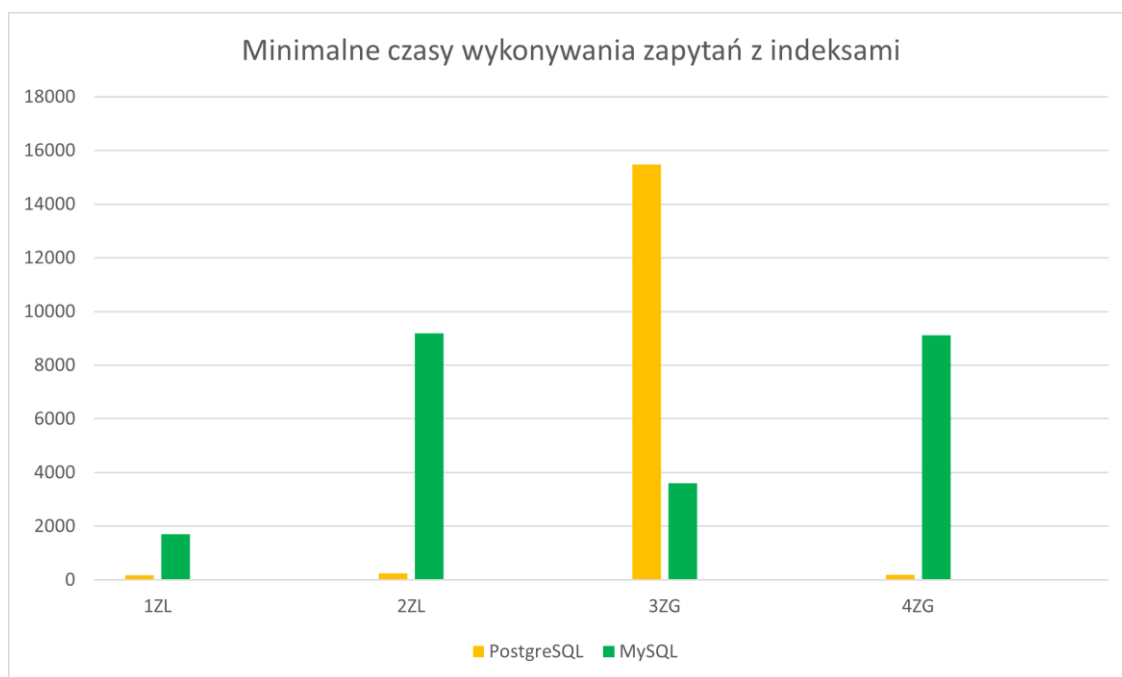
Tabela 1 Zestawienie wyników otrzymanych dla poszczególnych zapytań w poszczególnych systemach

	1ZL		2ZL		3ZG		4ZG	
BEZ INDEKSÓW	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
PostgreSQL	190	242,2	266	336,9	15650	15959,6	206	258,4
MySQL	3687	3718	11172	11322,2	4719	4755	11172	11234,6
Z INDEKSAMI								
PostgreSQL	177	241,5	247	300,1	15476	15728,8	190	252,4
MySQL	1703	1743,8	9188	9275,2	3594	3675,4	9110	9272

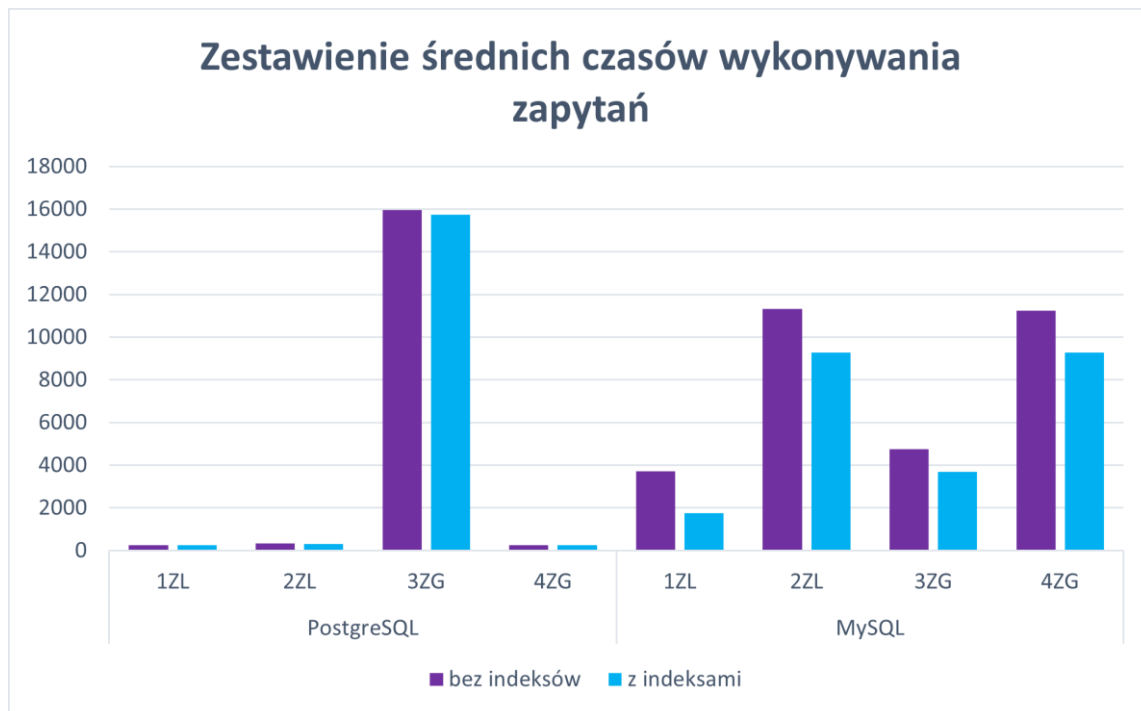
Aby ułatwić analizę wykonano graficzne zestawienie otrzymanych danych w postaci wykresów (Rys.1-3).



Rys. 1



Rys. 2



Rys. 3

6. Wnioski

Na podstawie przeprowadzonych testów oraz analizie ich wyników można wyciągnąć następujące wnioski:

- W systemie PostgreSQL różnice w czasach wykonania zapytań z indeksami i bez są nieznaczne, natomiast w MySQL bardzo wyraźne.
- Indeksowanie skraca czas wykonywania zapytań, ale nie w każdym oprogramowaniu znacząco.
- W większości przypadków PostgreSQL przetwarza zapytania znacznie szybciej niż MySQL.
- Postać znormalizowana tabel wydłuża czas wykonywania zapytań, z wyjątkiem zapytania 3ZG w PostgreSQL gdzie dla zdenormalizowanej postaci wykonuje się ono znacznie dłużej.