



Ćwiczenie 9

Julia Tulin, numer albumu: 410258

1. Wstęp

Celem poniżej opisanych testów było porównanie wydajności złączeń oraz zapytań zagnieżdżonych na dużej liczbie danych, które uprzednio stworzono poprzez połączenie danych z tabeli geochronologicznej z danymi z tabeli Milion.

Ćwiczenie to zostało wykonane na podstawie artykułu autorstwa prof. nadzw. dr hab. inż. Adama Piórkowskiego oraz Łukasza Jajeńnicy pt. ” WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH”

2. Parametry komputera:

CPU: 11th Gen Intel(R) Core(TM) i7-1195G7 @ 2.90GHz 2.92 GHz

RAM: 20,0 GB

SO: Windows 2011 Pro

SSD: NVMe Samsung SSD 980 1TB

System zarządzania bazami danych: - MySQL, wersja Community 8.0.33

- PostgreSQL 2023.1.2

3. Stworzenie bazy danych

Pierwszym krokiem było utworzenie bazy danych zawierającej tabele GeoEon, GeoEra, GeoOkres, GeoPietro i wypełnienie ich odpowiednimi danymi.

Następnie utworzono tabelę Geotabela używając komendy:

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

Do wykonania testów potrzebne były także tabele Dziesiec wypełniona liczbami 0-9 oraz utworzona na jej podstawie tabela Milion przechowująca liczby 0-999999 (trwało to 8610 ms)

4. Realizacja testów

- Zapytanie 1 (1 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn.

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON  
(mod(Milion.liczba,77)=(GeoTabela.id_pietro));
```

- Zapytanie 2 (2 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel.

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON
(mod(Milion.liczba,77)=GeoPietro.id_pietro) NATURAL JOIN GeoEpoka
NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

- 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 mod(Milion.liczba,77)=
(SELECT id_pietro FROM GeoTabela
WHERE mod(Milion.liczba,77)=(id_pietro));
```

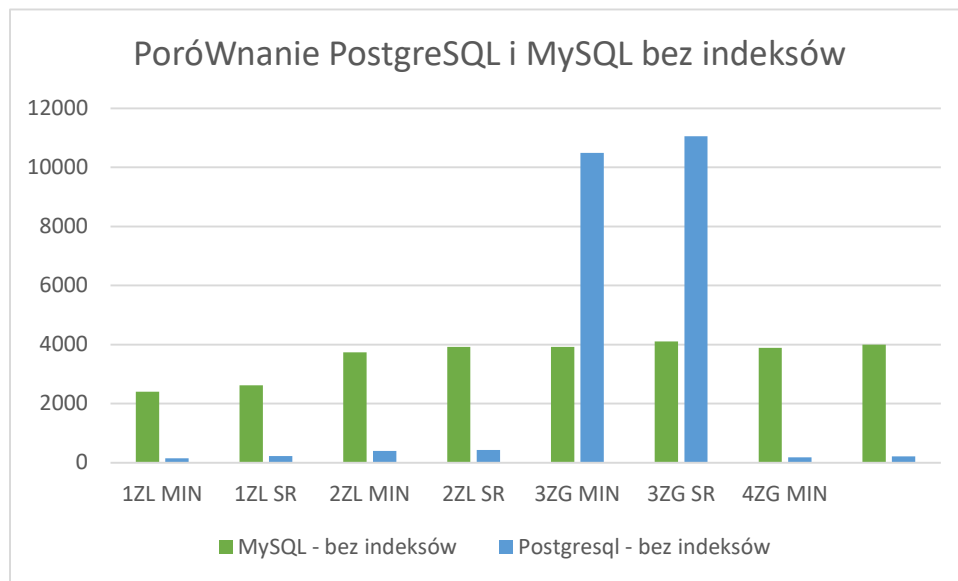
- Zapytanie 4 (4 ZG), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych.

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77)
IN (SELECT GeoPietro.id_pietro FROM GeoPietro
NATURAL JOIN GeoEpoka NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon);
```

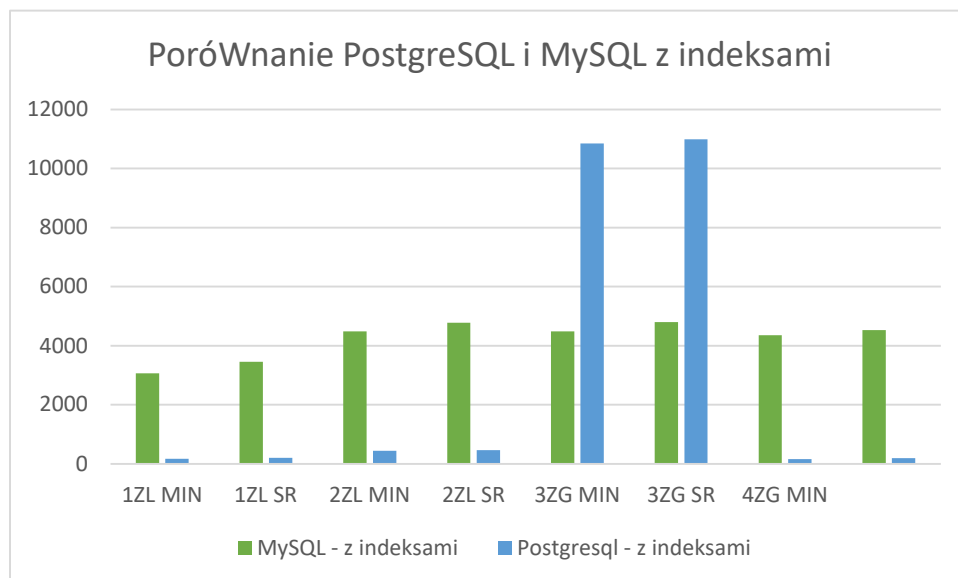
Czas wykonywania zapytań 1ZL, 2ZL, 3ZG, 4ZG [ms]								
	1ZL MIN	1ZL ŚR	2ZL MIN	2ZL ŚR	3ZG MIN	3ZG ŚR	4ZG MIN	4ZG ŚR
Bez indeksów								
MySQL	2406	2625	3734	3922	3922	4110	3891	4000
PostgreSQL	146	224	395	434	10496	11050	179	218
Z indeksami								
MySQL	3063	3453	4485	4782	4484	4797	4359	4531
PostgreSQL	167	200	446	463	10847	10992	165	187

Tabela 1. Wyniki testów uzyskane w MySQL i PostgreSQL.

5. Wykresy



Rysunek 1. Porównanie wyników testów uzyskanych w MySQL i PostgreSQL z indeksami na kluczach głównych.



Rysunek 2. Porównanie wyników testów uzyskanych w MySQL i PostgreSQL z indeksami na wszystkich kolumnach.

6. Wnioski

Porównując wyniki otrzymane w MySQL oraz w PostgreSQL można wysunąć następujące wnioski:

- W systemie MySQL czas wykonania wszystkich czterech zapytań oscyluje w granica od 2500 ms do 4500 ms. Z pierwszych dwóch zapytań nieznacznie szybsza okazuje się postać zdenormalizowana, natomiast z porównując trzecie i czwarte zapytanie widzimy że wykonują się one w bardzo podobnym czasie (3G wykonuje się minimalnie szybciej).
- W systemie MySQL utworzenie indeksów dla wszystkich kolumn tabel biorących udział w złączeniu wydłużyło czas wykonania jedynie o ok. 200 ms. W PostgreSQL także nastąpiła nieznaczna zmiana czasu wykonania.
- PostgreSQL czas wykonania się zapytań 1ZL, 2ZL i 4ZG jest zdecydowanie szybszy niż w MySQL. Zarówno bez indeksów jak i z indeksami nie przekracza 500 ms.
- W PostgreSQL zapytanie 3ZG wykonuje się znacznie dłużej niż pozostałe tak więc w tym przypadku lepsza jest postać znormalizowana 4ZG.

Analiza udowadnia, że nie zawsze normalizacja pozwala osiągnąć lepszą wydajność, jednak są przypadki (jak we wniosku 4) kiedy skraca ona czas wykonania nawet kilkadziesiąt razy.