



ĆWICZENIE 10

Analiza porównawcza wydajności złączeń i zagnieżdżeń dla
schematów znormalizowanych i zdenormalizowanych

Krzysztof Rojowski, nr. albumu: 412389
WGGIOŚ, Geoinformatyka

Wstęp

Celem niniejszego opracowania jest porównanie wydajności złączeń oraz zapytań zagnieżdżonych na dużej liczbie danych, które zostały uprzednio stworzone poprzez połączenie danych z tabeli geochronologicznej z danymi z tabeli *Milion*. Porównaniu podlega również wpływ indeksacji na szybkość wykonywania poleceń. W testach posłużono się systemami MySQL oraz PostgreSQL.

Konfiguracja sprzętowa i programowa

Wszystkie testy omówione w niniejszym opracowaniu odbyły się z wykorzystaniem komputera o następujących parametrach:

- CPU: 12th Gen Intel(R) Core(TM) i5-1235U 1.30 GHz,
- RAM: 16.0 GB
- OS: Windows 11 Home
- Dysk twardy: SSD 512GB

Jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

- PostgreSQL 16.3
- MySQL 8.0.37

Utworzenie baz danych

Na samym początku należało utworzyć bazę danych zawierającą tabele *GeoEon*, *GeoEra*, *GeoOkres*, *GeoPietro* oraz wypełnić je odpowiednimi danymi. Następnie została utworzona została tabela *Geotabela* za pomocą komendy:

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

W zapytaniach testowych łączono dane z tabeli geochronologicznej z syntetycznymi danymi o rozkładzie jednostajnym z tabeli *Milion*, wypełnionej kolejnymi liczbami naturalnymi od 0 do 999 999. Tabela *Milion* została utworzona na podstawie odpowiedniego autozłączenia tabeli *Dziesiec* wypełnionej liczbami od 0 do 9.

Realizacja testów wydajności

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

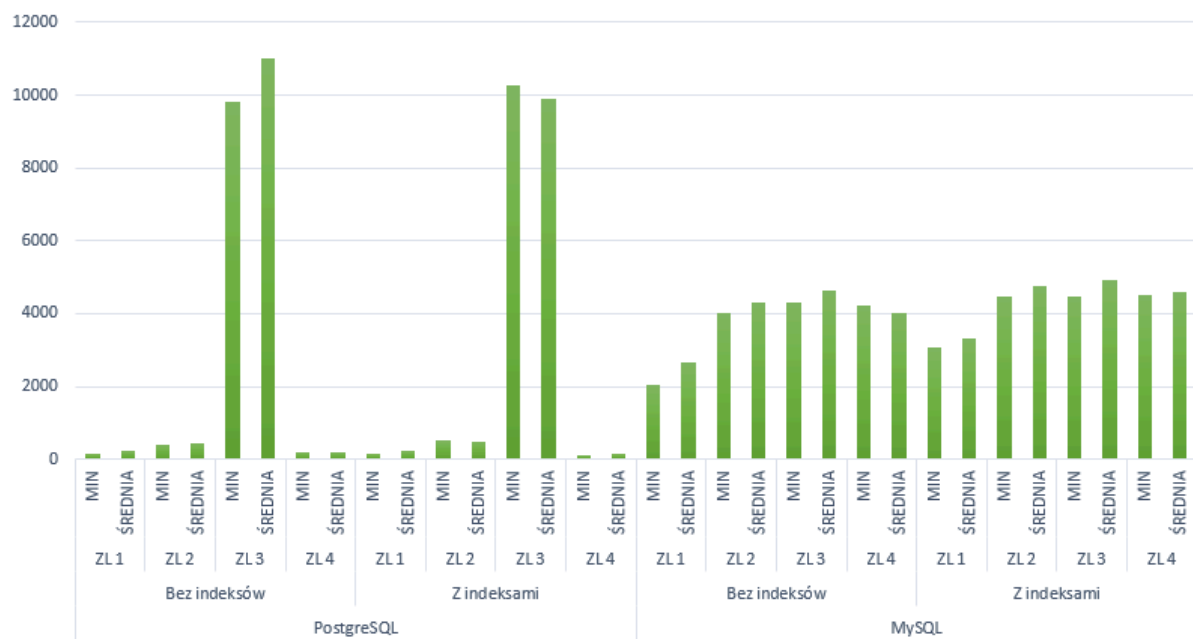
- **Zapytanie 4 (4 ZG)**, którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych:

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

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

	1 ZL		2 ZL		3 ZL		4 ZL	
Bez indeksów	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
MySQL	2041	2682	4013	4305	4314	4620	4214	4000
PostgreSQL	145	225	401	450	9808	11002	181	201
Z indeksami								
MySQL	3061	3310	4481	4745	4453	4942	4502	4592
PostgreSQL	159	240	515	490	10252	9910	130	162

Tabela 1. Wyniki testów uzyskane w MySQL i PostgreSQL



Wnioski

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

- W systemie MySQL czas wykonania wszystkich czterech zapytań zarówno z indeksami jak i bez indeksów oscyluje w granica od 2000 ms do 5000 ms. Z pierwszych dwóch zapytań szybsza okazuje się postać zdenormalizowana, natomiast z porównując trzecie i czwarte zapytanie widzimy że trzecie zapytanie wykonuje się szybciej o około 300ms.
- PostgreSQL czas wykonania się zapytań 1ZL, 2ZL i 4ZG jest zdecydowanie szybszy niż w MySQL. Zarówno bez indeksów jak i z indeksami nie przekracza 515 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 w każdym przypadku normalizacja pozwala osiągnąć lepszą wydajność. Są jednak przypadki (wniosek 3) kiedy normalizacja skraca czas wykonania.