

Kraków, 25.05.2023r.



Bazy Danych

**Ćwiczenie 9** – Badanie wydajności złączy i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych

Piotr Miniuk, nr albumu: 400467

## 1. Wstęp

Celem analizy było zbadanie i porównanie wydajności kwerend bazujących na złączeniach i zagnieżdżeniach dla tabeli geologicznej. Analizę przeprowadzono dla systemu bazodanowego opierającego się o koncept relacyjnych baz danych PostgreSQL.

## 2. Konfiguracja sprzętowa i programowa

COMPUTER:

- CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- GPU: NVIDIA GeForce GTX 1650
- SSD: INTEL SSDPEKNW512G8
- RAM: 8,00 GB
- OS: Windows 10 Pro

TOOLS: PostgreSQL 15.3 for Windows x86-64

IDE: pgAdmin4

## 3. Zapytania testowe

W celu przeprowadzenia testów stworzono tabelę Dziesięć oraz tabelę Milion zawierającą syntetyczne dane o jednorodnym rozkładzie od 0 do 999 999. Do wykonania testów użyto czterech zapytań oznaczonych jako 1ZL, 2ZL, 3ZG, 4ZG.

- 3.1. Zapytanie 1 (1ZL), 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 geol.Tabela ON  
(mod(Milion.liczba, 75) = (geol.Tabela.ID_Pietro));
```

- 3.2. Zapytanie 2 (2ZL), 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 geol.Pietro ON  
(mod(Milion.liczba, 75) = geol.Pietro.ID_Pietro) NATURAL JOIN geol.Epoka  
NATURAL JOIN geol.Okres NATURAL JOIN geol.Era NATURAL JOIN geol.Eon;
```

- 3.3. Zapytanie 3 (3ZG), 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, 75) = (SELECT  
ID_Pietro FROM geol.Tabela WHERE mod(Milion.liczba, 75) = (ID_Pietro));
```

- 3.4. Zapytanie 4 (4ZG), 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, 75) = (SELECT  
geol.Pietro.ID_Pietro FROM geol.Pietro NATURAL JOIN geol.Epoka NATURAL  
JOIN geol.Okres NATURAL JOIN geol.Era NATURAL JOIN geol.Eon);
```

## 4. Wyniki testów

Dla PostgreSQL wykonano pomiary wykonywania zapytania z indeksem jak i bez indeksu. Wykonano ręcznie 5 zapytań. Wszystkie wyniki zostały umieszczone w *Tabeli 1.* oraz *Tabeli 2.*

Bez indeksowania						
	Czas (ms)					średnia
Zapytanie 1	284	287	332	257	288	289,6
Zapytanie 2	217	228	175	127	141	177,6
Zapytanie 3	9819	10683	9770	9781	9999	10010,4
Zapytanie 4	191	149	199	179	248	193,2

Tabela 1. Wyniki pomiarów zapytań bez indeksowania

Z indeksowaniem						
	czas (ms)					średnia
Zapytanie 1	304	261	292	280	268	281
Zapytanie 2	174	226	173	179	164	183,2
Zapytanie 3	11410	11139	11569	11526	11466	11422
Zapytanie 4	167	186	187	219	195	190,8

Tabela 2. Wyniki pomiarów zapytań z indeksowaniem

## 5. Wnioski

Na podstawie powyższych wyników można stwierdzić, że indeksowanie wydłużyło czas złożonego zapytania 3, jednak nie wpłynęło to istotnie na pozostałe zapytania (niemal identyczne wyniki). Ponadto normalizacja w większości przypadków prowadzi do spadku wydajności, ale za to pozwala na łatwe przechowywanie danych w zrozumiałym sposób, zmniejsza szanse na wystąpienie błędów oraz porządkuje dane.