

WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŹDZEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

Dominika Szarek, nr 402890

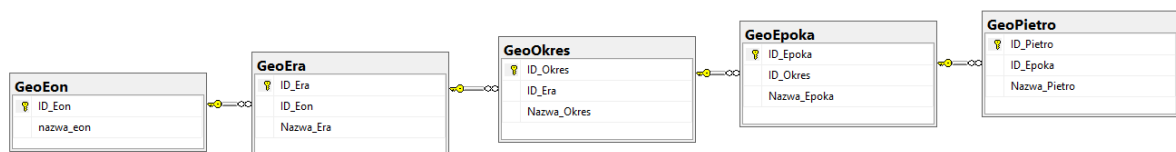
GEOINFORMATYKA, 2 rok

Akademia Górniczo – Hutnicza im. Stanisława Staszica

1. Schemat bazy danych

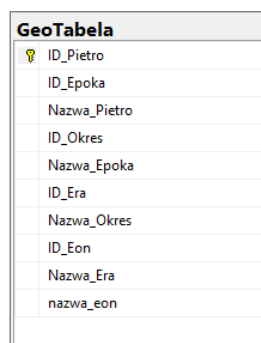
Porównanie wydajności złączeń i zagnieźdżeń dla schematów znormalizowanych i zdenormalizowanych przeprowadzono za pomocą tabeli stratygraficznej, która obrazuje przebieg historii Ziemi na podstawie następstwa procesów i warstw skalnych. W opracowaniu skupiono się na konstrukcji tabeli w dwóch przypadkach:

- Schemacie znormalizowanym



Rys. 1. Znormalizowany schemat tabeli stratygraficznej

- Schemacie zdenormalizowanym

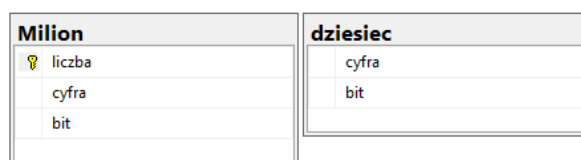


Rys 2. Zdenormalizowany schemat tabeli stratygraficznej

Testy przeprowadzono na tabelach o dużej liczbie danych. W tym celu powstała tabela Milion, wypełniona kolejnymi liczbami naturalnymi od 0 do 999 999, której dane łączono z danymi tabeli stratygraficznej.

Tabelę utworzono na podstawie złączenia tabeli Dziesięć wypełnionej liczbami od 0 do 9.

```
INSERT INTO Milion SELECT a1.cyfra + 10*a2.cyfra + 100*a3.cyfra +  
1000*a4.cyfra  
+ 10000*a5.cyfra + 100000*a6.cyfra AS liczba , a1.cyfra AS cyfra,  
a1.bit AS bit  
FROM dziesięć a1, dziesięć a2, dziesięć a3, dziesięć a4, dziesięć a5,  
dziesięć a6
```



Rys 3. Schemat tabeli Milion i Dziesięć

2. Konfiguracja sprzętowa i programowa

Testy zostały wykonane na komputerze o następujących parametrach:

- CPU: Intel(R) Core(TM) i7-4600U CPU 2.10GHz
- RAM: Pamięć DDR3 12.0 GB
- SSD: INTEL SSDSC2BF240A4L
- S.O.: Windows 10 Pro

Jako systemy zarządzania bazami danych wybrano oprogramowania wolno dostępne:

- Microsoft SQL Server Management Studio, wersja 15.0.18369.0
- PostgreSQL, wersja 13.3.2

3. Kryteria testów

W teście wykonano szereg zapytań sprawdzających wydajność złączeń i zagnieżdżeń z tabelą geochronologiczną w wersji zdenormalizowanej i znormalizowanej. Zapytania były wykonywane na tabelach o dużej liczbie danych. Procedura testowa obejmowała zapytania bez nałożonych indeksów na kolumny danych. W późniejszym etapie natomiast nałożono indeksy na kolumny biorące udział w złączeniu.

- Zapytanie 1 (1 ZL) – łączy tabelę syntetyczną miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej :

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela  
ON (Milion.liczba%68)=(GeoTabela.ID_Pietro)
```

- Zapytanie 2 (2 ZL) – łączy tabelę syntetyczną miliona wyników z tabelą geochronologiczną w postaci znormalizowanej w postaci złączenia pięciu tabel:

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro gpi  
ON (Milion.liczba%68)=gpi.ID_Pietro  
INNER JOIN GeoEpoka ep ON gpi.ID_Epoka = ep.ID_Epoka  
INNER JOIN GeoOkres okr ON okr.ID_Okres = ep.ID_Okres  
INNER JOIN GeoEra er ON er.ID_Era = okr.ID_Era
```

- Zapytanie 3 (3 ZG) – łączy tabelę syntetyczną miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, złączenie jest wykonywane poprzez zagnieżdżenie skorelowane:

```
SELECT COUNT(*) FROM Milion WHERE Milion.liczba%68=  
(SELECT id_pietro FROM GeoTabela WHERE  
Milion.liczba%68=(ID_Pietro));
```

- Zapytanie 4 (4 ZG) – łączy tabelę syntetyczną miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, złączenie wykonywane jest poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych:

```
SELECT COUNT(*) FROM Milion WHERE (Milion.liczba%68) =
(SELECT gpi.ID_Pietro FROM GeoPietro gpi
INNER JOIN GeoEpoka ep ON gpi.ID_Epoka = ep.ID_Epoka
INNER JOIN GeoOkres okr ON ep.ID_Okres = okr.ID_Okres
INNER JOIN GeoEra er ON okr.ID_Era = er.ID_Era
INNER JOIN GeoEon eo ON er.ID_Eon = eo.ID_Eon
WHERE Milion.liczba%68 =(ID_Pietro))
```

4. Wyniki testów

| | 1 ZL | | 2ZL | | 3ZL | | 4ZL | |
|---------------------|------|-----|------|------|-------|--------|-------|-------|
| BEZ INDEKSÓW | MIN | ŚR | MIN | ŚR | MIN | ŚR | MIN | ŚR |
| MSSQL Server | 126 | 141 | 117 | 142 | 5843 | 4049 | 74223 | 77185 |
| PostgreSQL | 636 | 733 | 1410 | 1562 | 20534 | 216989 | 22742 | 26368 |
| | | | | | | | | |
| Z INDEKSAMI | MIN | ŚR | MIN | ŚR | MIN | ŚR | MIN | ŚR |
| MSSQL Server | 93 | 104 | 100 | 122 | 93 | 117 | 6936 | 7100 |
| PostgreSQL | 433 | 498 | 769 | 908 | 16578 | 17660 | 45288 | 47731 |

Każdy test przeprowadzono kilkakrotnie. Czas wykonania zapytań przedstawiony jest w milisekundach [ms].

5. Wnioski

Na podstawie wyników możemy zauważyć, że tabela w postaci zdenormalizowanej w większości przypadków jest wydajniejsza, o czym świadczą porównane wyniki zapytania 1 ZL do reszty zapytań. Zagnieżdżenia skorelowane charakteryzują się zdecydowanie wolniejszym czasem wykonania niż złączenia, co obserwujemy na podstawie zapytań 3 ZG i 4 ZG porównując je do dwóch poprzednich.

W systemie MSSQL Server Management Studio użycie indeksów przyspieszyło czas wykonania zapytania w każdym badanym przypadku. Podobnie sytuacja prezentuje się w systemie PostgreSQL, wyłączając zapytanie 4 ZG, gdzie indeksy znacząco wydłużyły czas zapytania.

Bibliografia:

- <https://www.yumpu.com/xx/document/read/30346243/pobierz-plik-referatu>