AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Badanie wydajności złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych

Sprawozdanie z ćwiczenia 9

Łukasz Firek

Wstęp:

Celem ćwiczenia jest zbadanie wydajności złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych w programie do zarządzania bazami danych PostgreSQL w wersji 15.3 oraz MySQL 8.0.

Opis ćwiczenia:

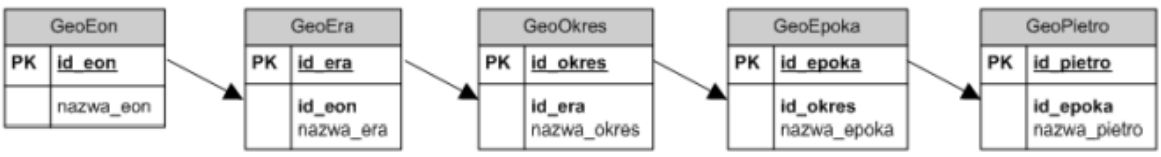
Poniższa tabela obrazuje przebieg historii Ziemi z podziałem na ery, okresy, epoki oraz zachodzące na niej procesy.

Tabela 1

Tabela geochronologiczna

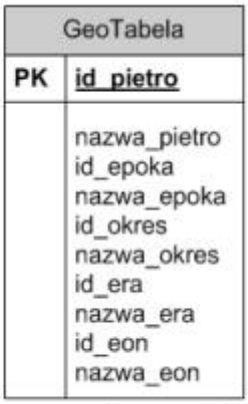
Wiek (mln lat)	Eon	Era	Okres		Epoka
0,010	FANEROZOIK	Kenzoik	Czwar-torząd		Halocen
1,8					Plejstocen
22,5			Trzeciorząd	Neo-gen	Pliocen
					Miocen
65				Paleogen	Oligocen
					Eocen
					Paleocen
140		Mezozoik	Kreda		Górna
					Dolna
195			Jura		Górna
					Środkowa
					Dolna
230			Trias		Górna
					Środkowa
					Dolna
280		Paleozoik	Perm		Górny
					Dolny
345			Karbon		Górny
					Dolny
395			Dewon		Górny
					Środkowy
					Dolny

Z powyższej tabeli należało stworzyć schemat znormalizowany, który był podzielony na jednostki geochronologiczne w tym eon, erę, okres, epokę oraz piętra (które niestety nie mogły zostać zobrazowane na powyższej tabeli ze względu na obszerność przedstawionych danych).



Rys. 1. Znormalizowany schemat tabeli geochronologicznej

Na podstawie znormalizowanego schematu tabeli geochronologicznej trzeba było stworzyć jej zdenormalizowany schemat o nazwie GeoTabela, która łączyła wszystkie powyższe schematy w jedną tabele.

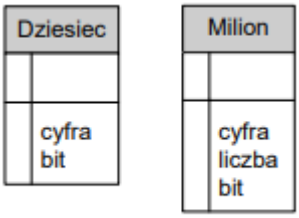


Rys. 2. Zdenormalizowany schemat tabeli geochronologicznej

Fragment kodu pozwalający na stworzenie schematu zdenormalizowanego z danych tabel, które były znormalizowane.

```
--nieznaturalizowana tabela
CREATE TABLE geo.GeoTabela AS (SELECT * FROM geo.geopietro NATURAL JOIN geo.geoepoka NATURAL
JOIN geo.geookres NATURAL JOIN geo.geoera NATURAL JOIN geo.geoeon );
```

Na potrzeby ćwiczenia należało stworzyć dodatkowe tabele Dziesiec oraz Milion, które były wypełnione danymi, między innymi liczbami od 0 do 999 999.



Rys. 3. Schemat tabel Dziesiec i Milion

Poniższe zapytania pozwalały na sprawdzenie wydajności powyższych schematów.

```sql
--Zapytanie 1 (1 ZL)
SELECT COUNT(*) FROM Milion INNER JOIN geo.GeoTabela ON
(mod(Milion.liczba,77)=(GeoTabela.id_pietro));

--Zapytanie 2 (2 ZL)
SELECT COUNT(*) FROM Milion INNER JOIN geo.GeoPietro ON
(mod(Milion.liczba,77)=geo.GeoPietro.id_pietro) NATURAL JOIN geo.GeoEpoka NATURAL JOIN
geo.GeoOkres NATURAL JOIN geo.GeoEra NATURAL JOIN geo.GeoEon;

--Zapytanie 3 (3 ZG)
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77)=
(SELECT id_pietro FROM geo.GeoTabela WHERE mod(Milion.liczba,68)=(id_pietro));

--Zapytanie 4 (4 ZG)
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77) in
(SELECT geo.GeoPietro.id_pietro FROM geo.GeoPietro NATURAL JOIN geo.GeoEpoka
    NATURAL JOIN geo.GeoOkres NATURAL JOIN geo.GeoEra NATURAL JOIN geo.GeoEon);
```

Specyfikacja urządzenia:

- CPU: Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz 2.90 GHz

- RAM: Pamięć 16,0 GB

- SSD: ADATA 250GB

- SO: Windows 10 Home

- PostgreSQL-15.3

-MySQL 8.0

Wyniki testów:

	1 ZL		2 ZL		3 ZL		4 ZL	
PostgreSQL	**MIN**	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**
Bez indeksów [ms]	167	172	520	527	7126	7154	160	165
Z indeksami [ms]	167	169	265	267	7130	7141	160	162

Tabela przedstawia wartości minimalne oraz średnią czasu potrzebnego na wykonanie powyższych zapytań (1 ZL, 2 ZL, 3 ZL, 4ZL) w milisekundach dla wartości z indeksami oraz bez nich.

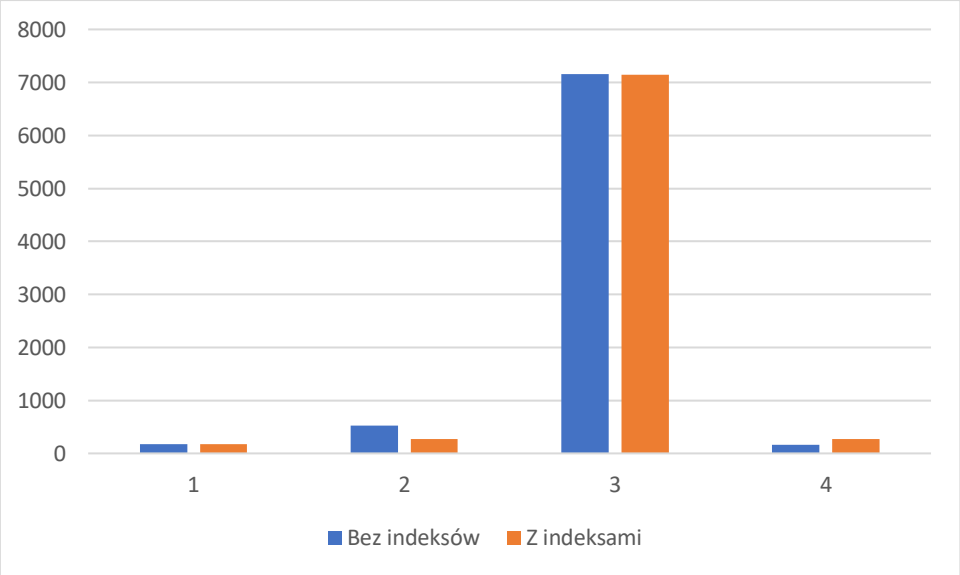


Tabela przedstawiająca zależności w średnim czasie [ms] wykonywania danych zapytań dla PostgreSQL.

	1 ZL		2 ZL		3 ZL		4 ZL	
MySQL	**MIN**	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**
Bez indeksów [ms]	1588	1603	652	658	2429	2446	651	657
Z indeksami [ms]	820	852	1319	1352	2176	2186	1308	1317

Tabela przedstawia wartości minimalne oraz średnią czasu potrzebnego na wykonanie powyższych zapytań (1 ZL, 2 ZL, 3 ZL, 4ZL) w milisekundach dla wartości z indeksami oraz bez nich w MySQL.

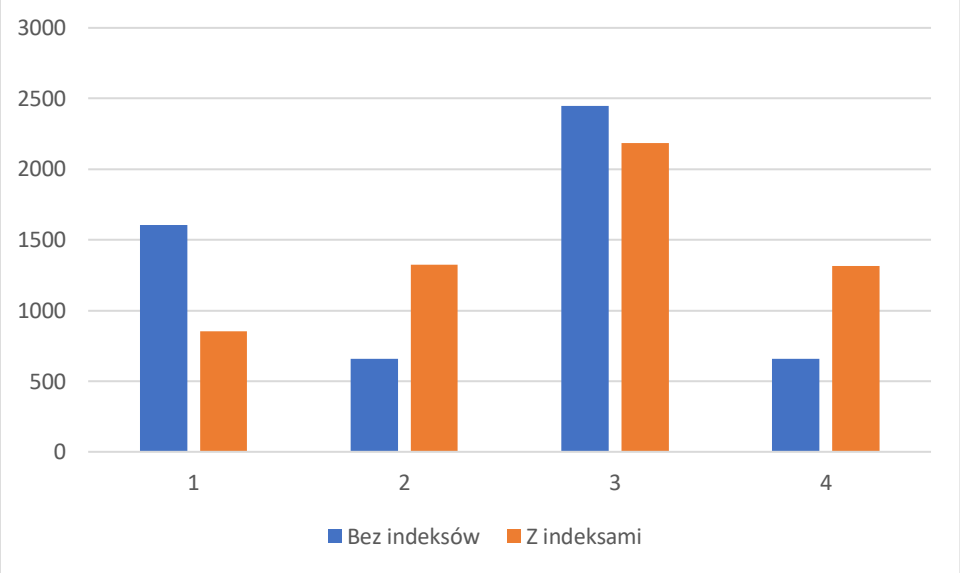


Tabela przedstawiająca zależności w średnim czasie [ms] wykonywania danych zapytań dla MySQL

Wnioski:

Podsumowując wyniki czasowe wykonywania danych zapytań w programie do zarządzania bazami danych PostgreSQL można dojść do wniosku, że indeksowanie znacząco nie wpływa na działanie i wydajność wykonywania danych zapytań. Bardziej znacząca różnica występuje jedynie w przypadku zapytania 2, gdzie średnia oraz wartość minimalna jest dwa razy większa dla wartości bez indeksowania niż dla tych z indeksowaniem. Natomiast w programie MySQL nie można zauważyć tak bardzo kolosalnej różnicy jak w poprzednim programie dla wykonywania zadania 3 natomiast, widać różnice ze względu na używanie danych z indeksem oraz bez niego. Dla niektórych zadań brak indeksowania pomaga przyspieszyć proces, ale dla innych ma działanie kompletnie odwrotne. Różnice między indeksowaniem, a jego brakiem oscylują w granicach dwukrotności w zależności od zadania. Porównując ze sobą działanie obu programów można dojść do wniosku, że program PostgreSQL lepiej poradził sobie z wykonywaniem powyższych zadań od programu MySQL.

Bibliografia:

Jajeśnica Ł., Piórkowoski A., WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH, Akademia Górniczo – Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej; Studia Informatica Vol. 31, No. 2A, Kraków 2010, s. 445÷456.