

Bazy danych ćw. 9 - projekt

Wykonanie: Dawid Olbrycht gr.2 284031

1. Cel projektu

Celem projektu było sprawdzenie wydajności złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych. Sprawdzenie to wykonano na specjalnie przygotowanych tabelach przedstawiających tabelę stratygraficzną. Wybrana przeze mnie tabela posiadała 98 pięter.

2. Parametry sprzętu i programu

Przy wykonywaniu pracy korzystałem z programu MySQL Community Server 8.0.23.

Pracowałem na komputerze z podanymi niżej podzespołami:

- Procesor Intel Core i7-7700HQ 2.8GHz 4-rdzenie
- Pamięć RAM 16 GB
- HDD Toshiba P300 2TB SATA III
- System operacyjny : Microsoft Windows 10 Home

3. Testy

Wykonałem szereg zapytań sprawdzających wydajność złączeń i zagnieżdżeń dla tabeli geochronologicznej w wersjach znormalizowanej i zdenormalizowanej. Zapytania te wykonałem w 2 etapach:

1. Dla tabel bez nałożonych indeksów na kolumnach
2. Dla tabel z nałożonymi indeksami

Zasadniczym celem testów była ocena wpływu normalizacji na zapytania złożone –złączenia i zagnieżdżenia (skorelowane) [14]. W tym celu zaproponowano cztery zapytania:

- **1 ZL**, którego celem było złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci ZDENORMALIZOWANEJ

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON (mod(Milion.liczba,98) =  
(GeoTabela.id_pietro));
```

- **2 ZL**, którego celem było złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci ZNORMALIZOWANEJ reprezentowanej przez złączenia 5 tabel

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON (mod(Milion.liczba,98)=GeoPietro.id_pietro)  
NATURAL JOIN GeoEpoka NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

- **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,98)= (SELECT id_pietro FROM GeoTabela WHERE  
mod(Milion.liczba,98)=(id_pietro));
```

- **4 ZG**, którego celem jest złączenie syntetycznej tablicy milion 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,98) IN (SELECT GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

Przy czwartym zapytaniu byłem zmuszony zmienić znak „=” na IN ponieważ w przeciwnym wypadku zapytanie nie chciało działać.

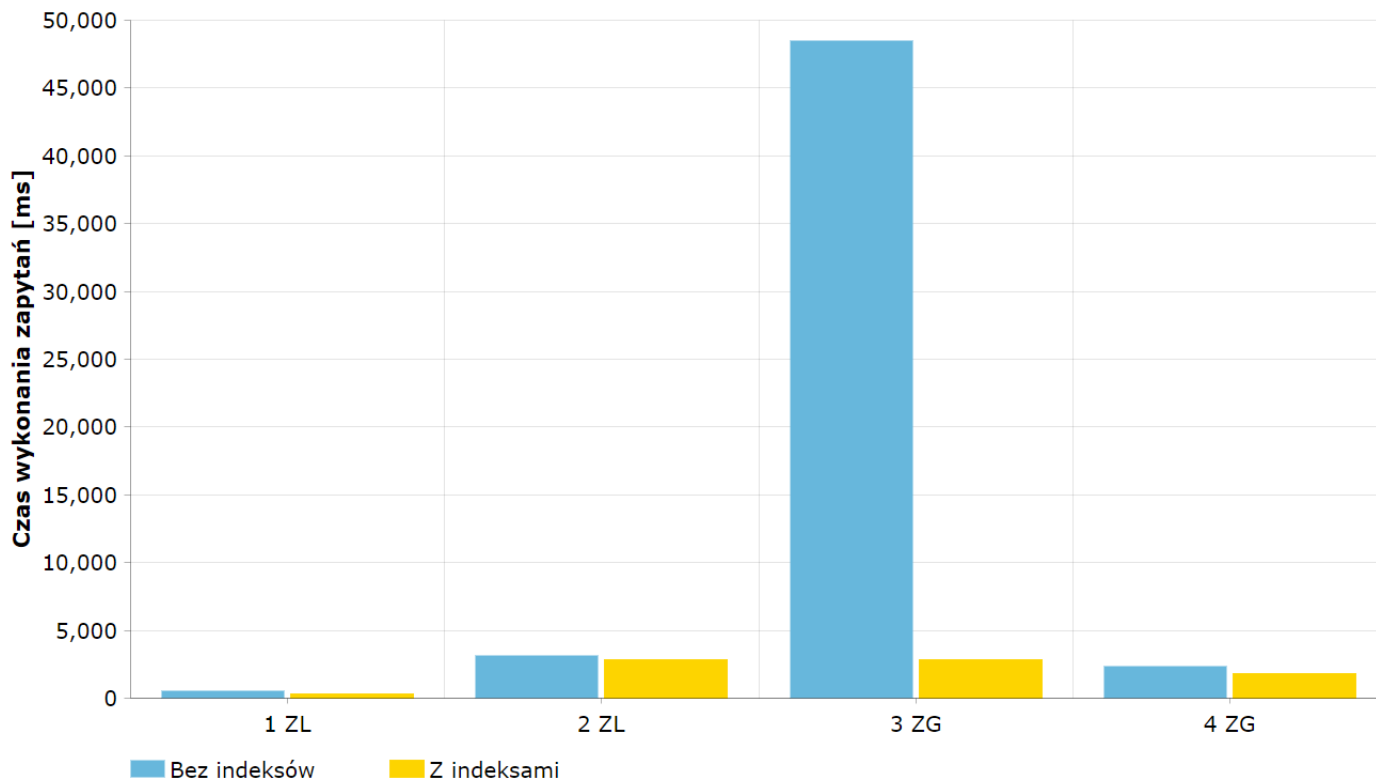
4. Wyniki

Każdy z testów przeprowadziłem 20 razy. Wyniki w każdej z prób były zgodne ze sobą. Z wszystkich 20 prób dla każdego testu wyciągnąłem średnią oraz wartość minimalną. Wartości średnich i wartości minimalnych umieściłem w tabeli 1.

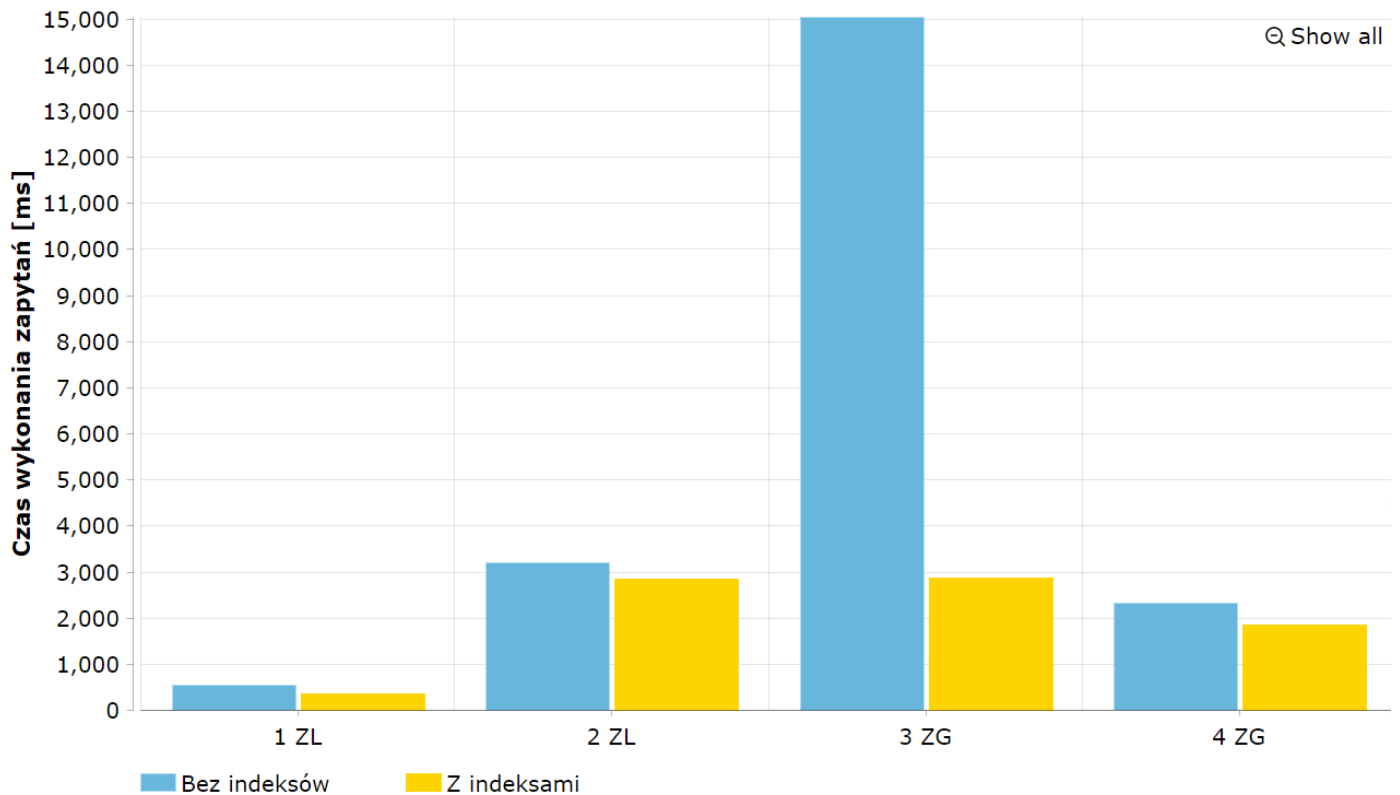
Tabela 1. Wyniki testów

	1 ZL		2 ZL		3 ZG		4 ZG	
	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
<u>Bez indeksów</u>	531	553,95	3031	3193,05	46922	48531,9	2219	2328,1
<u>Z indeksami</u>	328	353,8	2609	2836,8	2750	2874,2	1765	1851,45

W celu lepszej wizualizacji wyników sporządziłem 2 wykresy:



Rys1. Wyniki testów.



Rys.2 Wyniki testów w zbliżeniu na wartości niższe

5. Wnioski

Z otrzymanych wyników można wyciągnąć następujące wnioski:

- Użycie indeksów w systemie MySQL we wszystkich testach przyspiesza wykonanie zapytań dla zagnieżdżeń i złączeń.
- Dla złączeń postać zdenormalizowana jest wydajniejsza, natomiast dla zagnieżdżeń wydajniejszą postacią jest postać znormalizowana
- Najwięcej problemów systemowi MySQL sprawiło zagnieżdżenie skorelowane zdenormalizowanej wersji tabeli geochronologicznej