WYDZIAŁ GEOLOGII, GEOFIZYKI I OCHRONY ŚRODOWISKA

Katedra Geoinformatyki i Informatyki Stosowanej

„WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH”

w oparciu o artykuł autorstwa Łukasza Jajeśnicy oraz Adama Piórkowskiego

Geoinformatyka, semestr III

Autor:

Ewelina Szeliga, indeks: 406708

Data sporządzenia: 05.06.2022

1. CELE

Celem wykonywanych badań jest sprawdzenie wydajności w kategoriach złączeń i zagnieżdżeń, schematach znormalizowanych i zdenormalizowanych oraz indeksacji lub braku indeksów. Szukana jest odpowiedź na hipotezę „Czy wersja znormalizowana jest wolniejsza czy szybsza od wersji zdenormalizowanej?”

2. SPRZĘT I DANE

W badaniu jako dane przyjęto tabele z danymi geochronologicznymi oraz tabelę z liczbami od 0 do 999999 i osobną z liczbami od 0 do 9.

id_eon	id_era	id_okres	id_epoka	id_pietro	nazwa_pietro	nazwa_epoka	nazwa_okres	nazwa_era	nazwa_eon
1	1	3	6	19	roadian	dolny	perm	paleozoik	fanerozoik
1	1	3	7	20	wordian	górny	perm	paleozoik	fanerozoik
1	1	3	7	21	kapitanian	górny	perm	paleozoik	fanerozoik
1	1	3	7	22	wuchiapingian	górny	perm	paleozoik	fanerozoik
1	1	3	7	23	changhsingian	górny	perm	paleozoik	fanerozoik
1	2	4	8	24	induan	dolna	trias	mezozoik	fanerozoik
1	2	4	8	25	olenekian	dolna	trias	mezozoik	fanerozoik
1	2	4	9	26	anisian	środkowa	trias	mezozoik	fanerozoik
1	2	4	9	27	ladinian	środkowa	trias	mezozoik	fanerozoik

Rysunek 1.Fragment GeoTabeli ze wszystkimi danymi geochronologicznymi, którą stworzono z użyciem NATURAL JOIN.

Wszystkie testy omówione w niniejszym artykule wykonano na komputerze o następujących parametrach:

CPU: Intel(R) Core(TM) i7-1065G7 1.30GHz,

RAM: Pamięć DDR4 8 GB (533 MHZ),

SSD: KBG40ZNS512G NVMe KIOXIA 512GB,

S.O.: Microsoft Windows 10 Home

Oraz jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

PostgreSQL, wersja 14.2-1

MySQL, wersja Community Server 8.0.29

3. KRYTERIA TESTÓW

Wykonano poniższe zapytania na tabelach bez nałożonych indeksów, następnie te same zapytania uruchomiono po nałożeniu indeksów na wszystkie kolumny, które brały udział w złączeniu.

Zapytanie 1 ZL :

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON (mod(Milion.liczba,68)=(GeoTabela.id_pietro));
```

Zapytanie 2 ZL:

```
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 ZG:

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

Zapytanie 4 ZG:

```
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;
```

Nałożenie indeksów:

```
CREATE INDEX idx_id_eon ON geotabele.GeoEra(id_eon);

CREATE INDEX idx_id_era ON geotabele.GeoOkres(id_era);

CREATE INDEX idx_id_okres ON geotabele.GeoEpoka(id_okres);

CREATE INDEX idx_id_epoka ON geotabele.GeoPietro(id_epoka);

CREATE INDEX idx_id_pietr_geotabela ON geotabele.geotabela(id_pietro);
```

4 . WYNIKI TESTÓW

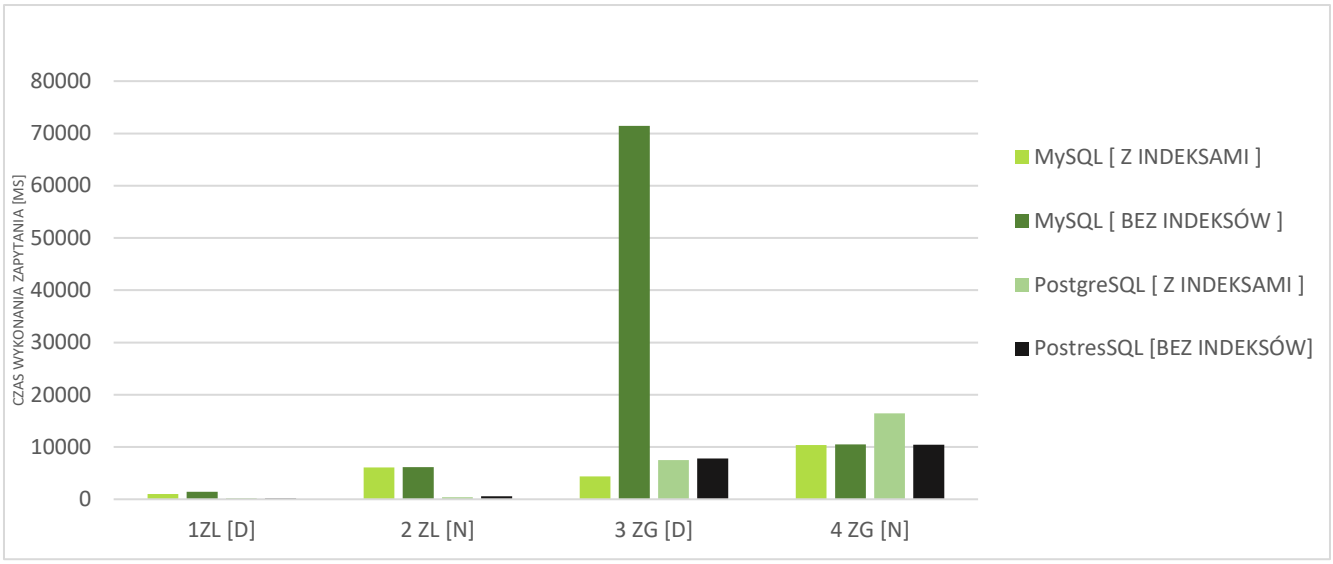
Każdy test został przeprowadzony 7-krotnie. Odrzucone zostały dwie skrajne wartości skąd wyciągnięto informacje o wartości minimalnej, maksymalnej i średniej szybkości wykonania zapytania. Wyniki testów zamieszczono w tabeli [1].

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

	1 ZL			2 ZL			3 ZG			4 ZG		
BEZ INDEKSÓW	MIN	ŚR	MAX	MIN	ŚR	MAX	MIN	ŚR	MAX	MIN	ŚR	MAX
PostgreSQL	157	175,4	189	572	595,2	615	7644	7822,6	8028	10021	10469,6	11053
MySQL	1344	1440,6	1516	6031	6140,8	6281	68828	71418,8	75969	10297	10481,2	10859
Z INDEKSAMI												
PostgreSQL	172	182	205	391	408,4	433	7392	7486,6	7574	16187	16443,3	16967
MySQL	937	1006	1046	6016	6100,4	6235	4032	4366	5157	9985	10374	10884

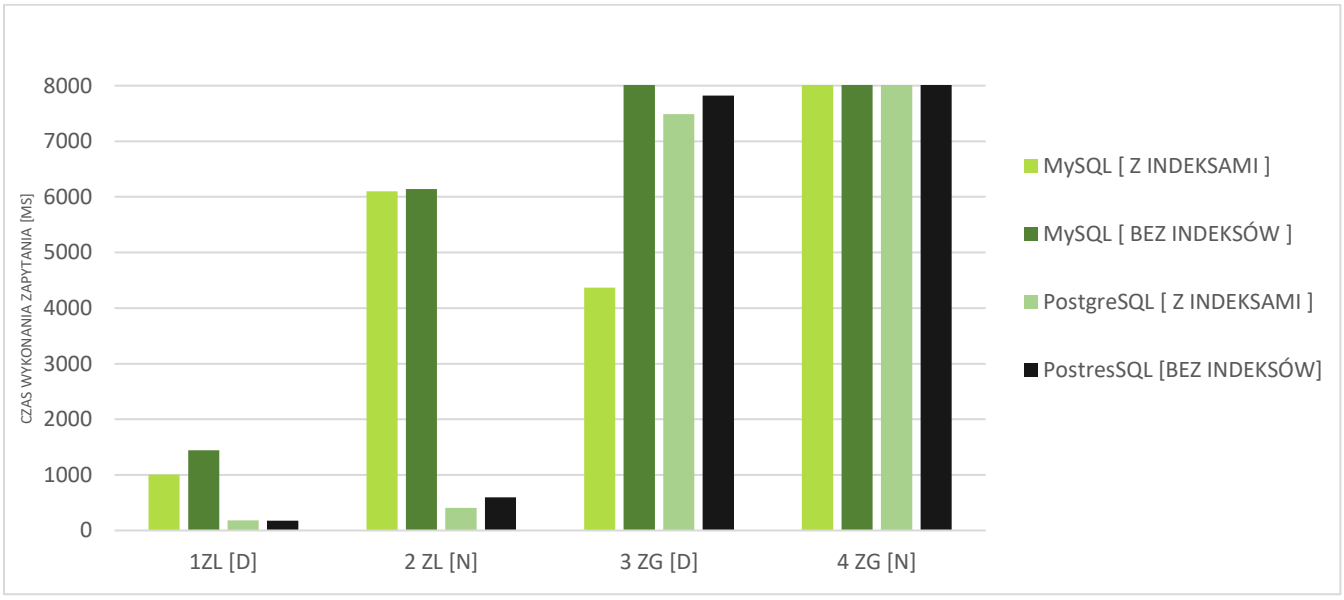
Na zielono zaznaczono szybsze średnie wyniki w danym zapytaniu w obrębie bazy. Jaśniejszy odcień zieleni odpowiada za bazę PostgreSQL, natomiast ciemniejszy MySQL.

Został stworzony również wykres słupkowy przedstawiający porównanie zapytań i czas ich wykonania w wersji do 80000 [ms] [Rysunek 2] oraz do 8000 [ms] [Rysunek 3] w celu lepszej czytelności dla niższych wyników.



Rysunek 2. Wyniki testów w wersji do 80 000 [ms]

Wyszczególniono również, które zapytania są zdenormalizowane (litera „D") i znormalizowane (litera „N") w celu wysunięcia wniosków dotyczących głównego zagadnienia o wydajności złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych.



Rysunek 3. Wyniki testów w wersji do 8000 [ms]

5. WNIOSKI

Opierając się na wynikach, które zostały uzyskane, można wyciągnąć następujące wnioski, dzieląc je na cztery kwestie:

- NORMALIZACJA
  - W większości przypadków, poza 3 zapytaniem zagnieżdżonym, zdenormalizowana postać jest szybsza w stosunku do niezdenormalizowanej.

- ZŁĄCZENIA/ZAGNIEŻDŻENIA
  - Złączenia wykonują się szybciej niż zagnieżdżenia.
  - Złączenia zdenormalizowane są szybsze względem złączeń niezdenormalizowanych
  - Podobnie jest dla zagnieżdżeń za wyjątkiem 3 ZG w MySQL bez indeksów.

- INDEKSACJA
  - 3 zapytanie w MySQL wykonuje się ponad 16 razy wolniej bez indeksów, niż przy zastosowaniu indeksacji.
  - Dodanie indeksów w większości przypadków zwiększyło szybkość, jednak wyjątkiem okazuje się PostgreSQL gdzie w dwóch przypadkach indeksacja spowalnia wykonywanie zapytań.
  - We wszystkich przypadkach MySQL z indeksami był szybszy niż bez indeksacji

- MySQL/PostgreSQL
  - Wyniki są w większości przypadków stabilne, co oznacza że średnie wartości policzone dla każdego zapytania są zbliżone w obrębie danej bazy.
  - Różnice pomiędzy min i max w szybkości zapytań w obrębie bazy również nie są znaczące.
  - Poza 3 ZG I 4 ZG z indeksami PostgreSQL był szybszy w wykonywaniu zapytań od MySQL.

Podsumowując indeksacja zwiększa w większości przypadków szybkość wykonywania zapytań, jednak dodanie normalizacji ma wpływ na zmniejszenie wydajności. Jednak koszt braku normalizacji jaki się ponosi, czasem może przewyższyć zalety szybszego wyświetlania wyników. Tracimy bowiem czytelność bazy, pojawiają się problemy z redundancją danych oraz anomalie usuwania, w której możemy tracić informacje.

5. LITERATURA

Badania zostały przeprowadzone w oparciu o artykuł „WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH” autorstwa: Łukasz Jajeśnica i Adam Piórkowski, STUDIA INFORMATICA 2010, Volume 31, Number 2A (89).