WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

Hubert Gołda

417 448

Geoinformatyka II rok

Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

1. Wprowadzenie

Bazy danych odgrywają kluczową rolę w nowoczesnych systemach informatycznych będąc fundamentem dla przechowywania, zarządzania oraz analizowania dużych zbiorów danych. W kontekście wydajności działania baz danych niezwykle istotnym aspektem jest optymalizacja. Wpływa ona bezpośrednio na szybkość wykonywania zapytań. Jej proces obejmuje szereg technik i strategii, które mają na celu usprawnienie działania bazy danych oraz zapewnienie szybkiego dostępu do informacji.

Jednym z podstawowych elementów optymalizacji baz danych jest normalizacja. Polega ona na podziale danych na tabele w taki sposób, aby zredukować redundancję oraz eliminować potencjalne anomalie podczas operacji na danych. Proces ten składa się z kilku etapów, znanych jako formy normalne, z których każda eliminuje określone problemy związane z redundancją i integralnością danych. Choć normalizacja pomaga w utrzymaniu porządku i spójności danych, może prowadzić do złożonych struktur tabel, które mogą wpłynąć na szybkość wykonywania zapytań, zwłaszcza gdy konieczne są częste złączenia między tabelami.

Złączenia, czyli operacje łączenia danych z różnych tabel na podstawie określonych warunków, są nieodłącznym elementem pracy z relacyjnymi bazami danych. Wydajność złączeń może być kluczowa dla szybkości działania systemu, szczególnie gdy pracujemy z dużymi zbiorami danych. Optymalizacja złączeń poprzez odpowiednie indeksowanie, wybór właściwych typów złączeń oraz minimalizację liczby niepotrzebnych operacji znacząco wpływa na czas odpowiedzi zapytań.

Innym czynnikiem odgrywającym ważną rolę w kontekście skomplikowanych operacji na danych są zagnieżdżenia. Są to zapytania zawierające inne zapytania. Choć zagnieżdżenia mogą być potężnym narzędziem umożliwiającym realizację złożonych zapytań w sposób czytelny i logiczny, ich nieodpowiednie użycie może prowadzić do znaczących spadków wydajności. Dlatego kluczowe jest umiejętne zarządzanie zagnieżdżeniami, w taki sposób aby unikać zbędnych obciążeń systemu.

Optymalizacja baz danych, normalizacja, złączenia oraz zagnieżdżenia stanowią fundamentalne aspekty wpływające na efektywność i szybkość wykonywania zapytań. Zrozumienie tych procesów oraz umiejętne ich zastosowanie jest niezbędne dla tworzenia wydajnych i niezawodnych systemów bazodanowych. W niniejszej pracy zbadany zostanie wpływ indeksowania i normalizacji na czas wykonywania zapytań, a także różnice między złączeniami a zagnieżdżeniami.

2. Tabela geochronologiczna

W eksperymencie wykorzystana została zmodyfikowana tabela geochronologiczna. Jej zadaniem jest zorganizowanie i zobrazowanie przebiegu historii Ziemi na podstawie następstwa procesów wraz z warstwami skalnymi. Ta tabela jest kluczowym narzędziem wykorzystywanym przez geologów oraz paleontologów, ponieważ precyzyjnie określa oficjalną terminologię okresów geologicznych w historii Ziemi, co zapobiega używaniu tych samych nazw w różnych kontekstach w publikacjach naukowych. W różnych opracowaniach mogą występować różne podziały stratygraficzne ze względu na trwające badania i wyznaczanie nowych standardowych profili (stratotypów).

W tabeli 2.1 przedstawiono bazową, obowiązującą wersję tabeli, jednak ze względu na obszerność i skomplikowanie ostateczna tabela została minimalnie uproszczona.

Tabela 2.1 Aktualnie obowiązująca tabela geochronologiczna, źródło [2]

Indywidualnie sporządzona tabela geochronologiczna zawiera pięć jednostek geochronologicznych: eon, erę, okres, epoki oraz piętra. Występują one odpowiednio w liczbie sztuk 1, 3, 8, 22 oraz 68.

3. Konstrukcja wymiaru geochronologicznego

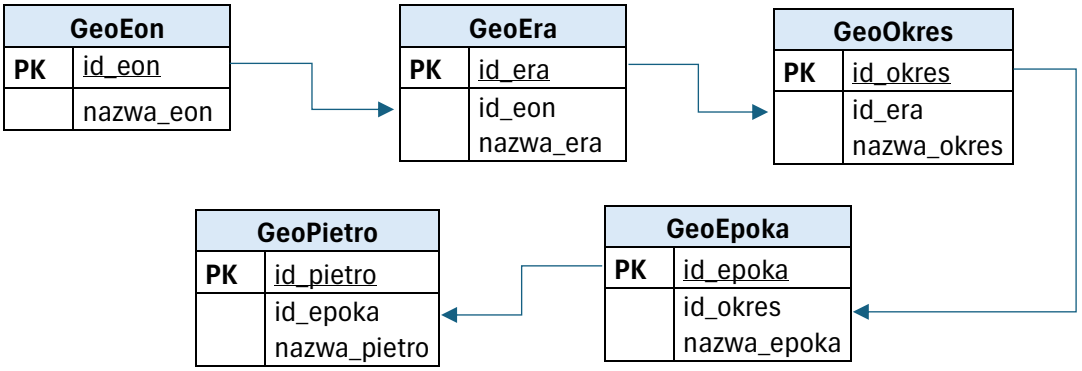
W tabeli geochronologicznej uwzględniono jednostki geochronologiczne, które mają wymiar czasowy (eon, era, okres, epoka i wiek), oraz odpowiadające im jednostki stratygraficzne. Te informacje stanowią wymiar, który jest obecny w wielu bazach danych geologicznych. W niniejszym opracowaniu skoncentrowano się na budowie tabeli geochronologicznej w dwóch przypadkach:

- schemacie znormalizowanym (płatka śniegu, rys. 3.1),
- schemacie zdenormalizowanym (schemat gwiazdy).

Formę zdenormalizowaną tabeli geochronologicznej osiągnięto tworząc jedną tabelę GeoTabela (rys. 3.2), zawierającą wszystkie dane z powyższych tabel. Dokonano tego za pomocą złączenia naturalnego, obejmującego wszystkie tabele tworzące hierarchię:

```
CREATE TABLE zdenormal.GeoTabela AS (SELECT * FROM
znormal.GeoPietro NATURAL JOIN znormal.GeoEpoka NATURAL JOIN
znormal.GeoOkres NATURAL JOIN znormal.GeoEra NATURAL JOIN
znormal.GeoEon);
```

Stworzenie tabeli GeoTabela pozwoliło na szybki dostęp do wszystkich danych z tabeli geochronologicznej za pomocą jednego prostego zapytania, co nie jest możliwe przy użyciu znormalizowanego schematu opisanego w punkcie pierwszym.



Rys. 3.1. Schematy znormalizowanych tabeli geochronologicznej.

GeoTabela	
PK	id_pietro
	nazwa_pietro id_epoka nazwa_epoka id_okres nazwa_okres id_era nazwa_era id_eon nazwa_eon

Rys. 3.2. Zdenormalizowany schemat tabeli geochronologicznej.

4. Testy wydajności

Podczas wykonywania testów największą uwagę położono na porównaniu wydajności złączeń oraz zapytań zagnieżdżonych wykonywanych. Były one wykonywane na tabelach z dużą ilością danych. Do testów użyto darmowej bazy danych PostgreSQL.

W zapytaniach testowych połączono dane z tabeli geochronologicznej z syntetycznymi danymi o rozkładzie jednostajnym z tabeli Milion, wypełnionej liczbami naturalnymi od 0 do 999 999. Tabela Milion została stworzona poprzez odpowiednie autozłączenie tabeli Dziesiec, zawierającej liczby od 0 do 9. Sposób stworzenia powyższych tabel:

```
DROP table Dziesiec;
CREATE TABLE Dziesiec (cyfra INT, bit INT);
INSERT INTO Dziesiec (cyfra, bit)
VALUES (0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (6, 0), (7,
0), (8, 0), (9, 0);

CREATE TABLE Milion(liczba int,cyfra int, bit int);
INSERT INTO Milion SELECT a1.cyfra +10* a2.cyfra +100*a3.cyfra +
1000*a4.cyfra + 10000*a5.cyfra + 10000*a6.cyfra AS liczba,
a1.cyfra AS cyfra, a1.bit AS bit FROM Dziesiec a1, Dziesiec a2,
Dziesiec a3, Dziesiec a4, Dziesiec a5, Dziesiec a6;
```

Dziesiec	
	cyfra bit

Milion	
	cyfra liczba bit

Rys. 4.1. Schematy tabel Dziesiec oraz Milion.

4.1. Konfiguracja sprzętowa i programowa

Testy wydajności opisane w tym eksperymencie zostały przeprowadzone na komputerze o następującej specyfikacji:

- CPU: Intel Core i7-11800h @2.3 GHz,
- RAM: Pamięć 16 GB (DDR4, 3200 MHz),
- SDD: 1 TB M.2 NVMe PCIe 3.0 SSD,
- S.O.: Windows 11, version 23H2.

Wybrany systemy zarządzania bazami danych:

- PostgreSQL, wersja 16.2-1.

Testy zostały wykonano kilkakrotnie.

4.2. Kryteria testów

W teście wykonano szereg zapytań sprawdzających wydajność złączeń i zagnieżdżeń z tabelą geochronologiczną w wersji zdenormalizowanej i znormalizowanej. Procedurę testową przeprowadzono w dwóch etapach:

- W pierwszy etapie testowane były zapytania bez nałożonych indeksów na kolumny danych (jedynymi indeksowanymi danymi były dane w kolumnach będących kluczami głównymi poszczególnych tabel),
- Podczas drugiego etapu nałożono indeksy na wszystkie kolumny biorące udział w złączeniu. Sposób indeksowania:

```
CREATE INDEX idx_GeoTabela ON zdenormal.GeoTabela (id_eon,
id_era, id_okres, id_epoka, id_pietro, nazwa_pietro,
nazwa_epoka, nazwa_okres, nazwa_era, nazwa_eon);
CREATE INDEX idxGeoEon ON znormal.GeoEon
(id_eon,nazwa_eon);
```

... itd. dla każdej wykorzystanej tabeli.

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

- **Zapytanie 1 (1 ZL)**, 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 zdenormal.GeoTabela
ON(mod(Milion.liczba,68)=(zdenormal.GeoTabela.id_pietro));
```

- **Zapytanie 2 (2 ZL)**, 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 znormal.GeoPietro ON
(mod(Milion.liczba,68)=znormal.GeoPietro.id_pietro)
NATURAL JOIN znormal.GeoEpoka NATURAL JOIN znormal.GeoOkres
NATURAL JOIN znormal.GeoEra NATURAL JOIN znormal.GeoEon;
```

- **Zapytanie 3 (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,68)=
(SELECT id_pietro FROM zdenormal.GeoTabela WHERE
mod(Milion.liczba,68)=(id_pietro));
```

- **Zapytanie 4 (4 ZG)**, 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, 68) IN (
    SELECT znormal.GeoPietro.id_pietro
    FROM znormal.GeoPietro
    NATURAL JOIN znormal.GeoEpoka
    NATURAL JOIN znormal.GeoOkres
    NATURAL JOIN znormal.GeoEra
    NATURAL JOIN znormal.GeoEon);
```

5. Wyniki testów

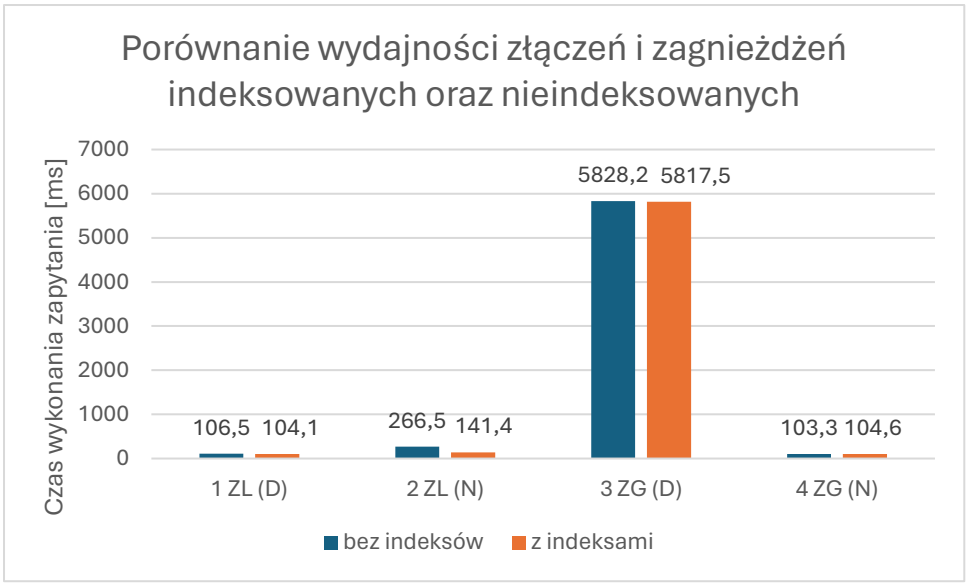
W ramach eksperymentu zostały przeprowadzone wielokrotne testy. Dla lepszego rezultatu skrajne czasy wykonania zapytań pominięto. Wyniki testów zamieszczono w tabeli 4.1 i 4.2 oraz na wykresach 41. i 4.2.

PostgreSQL						
zapytanie	**1 ZL**			**2 ZL**		
	średnia	**min**	**max**	**średnia**	**min**	**max**
bez indeksów [ms]	106,527	102,927	112,604	266,520	253,734	273,875
z indeksami [ms]	104,133	102,506	105,956	141,449	133,431	149,996
różnica [ms]	-2,393			-125,071		
różnica [%]	2,25%			46,93%		

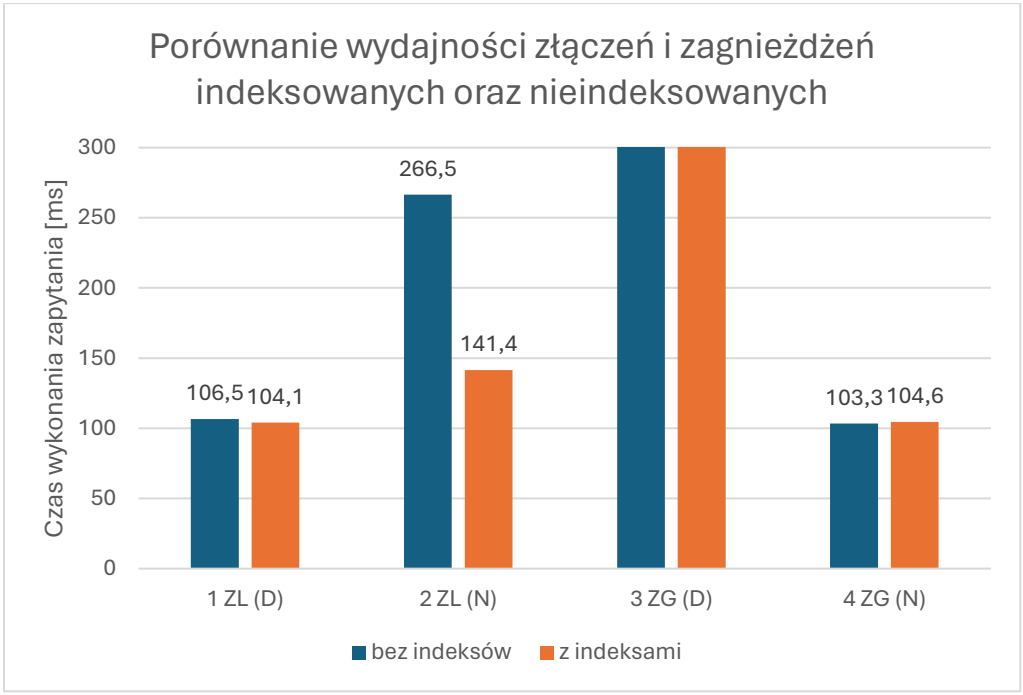
Tabela 4.1. Zestawienie czasów wykonania zapytań 1 ZL, 2 ZL.

PostgreSQL						
zapytanie	**3 ZG**			**4 ZG**		
	średnia	**min**	**max**	**średnia**	**min**	**max**
bez indeksów [ms]	5828,186	5685,113	6004,166	103,318	100,688	107,142
z indeksami [ms]	5817,463	5720,837	5953,596	104,583	101,182	109,384
różnica [ms]	-10,723			1,265		
różnica [%]	0,18%			-1,22%		

Tabela 4.2. Zestawienie czasów wykonania zapytań 3 ZG i 4 ZG.



Wykres 4.1. Porównanie wydajności złączeń i zagnieżdżeń indeksowanych oraz nieindeksowanych w PostgreSQL.



Wykres 4.2. Porównanie wydajności złączeń i zagnieżdżeń indeksowanych oraz nieindeksowanych w PostgreSQL z uwzględnieniem skali.

6. Wnioski

Na podstawie otrzymanych wyników można wyciągnąć następujące wnioski:

- W przypadku złączeń bez indeksowania postać zdenormalizowana okazała się być ponad 2,5 razy szybsza od postaci znormalizowanej.
- Różnica zmniejszyła się, kiedy zaindeksowano tabele. Wówczas postać zdenormalizowana była tylko niecałe 1,5 razy szybsza. Indeksowanie pozwoliło wykonać zapytanie złączone postaci znormalizowanej w czasie niemal o połowę krótszym.
- Była to zarazem jedyna widoczna różnica w czasie wykonywania zapytania. W innych sytuacjach indeksowanie było minimalnie szybsze, bądź nawet jak w przypadku zagnieżdżenia i postaci znormalizowanej trochę wolniejsze.
- W przypadku zagnieżdżeń postać znormalizowana zdeklasyfikowała postać zdenormalizowaną czasem wykonania. Zarówno indeksowana jak i nieindeksowana postać znormalizowana wykonała się niemal 60 razy szybciej od postaci zdenormalizowanej. Był to jedyny przypadek aż tak długiego czasu wykonywania się zapytania w tym eksperymencie.
- Na podstawie przeprowadzonego doświadczenia nie jest możliwe jednoznaczne stwierdzenie, która z postaci jest szybsza. Ostateczny wynik może zależeć od sposobu użycia danej metody.
- Indeksowanie tabel nie gwarantuje krótszego czasu wykonania zapytania, jednakże bardziej prawdopodobne jest, iż przyspieszy pracę, a nie zaszkodzi.

Podsumowując, ciężko ocenić pozytywny bądź negatywny wpływ normalizacji oraz indeksowania tabel na czas wykonywania zapytań. Oczywistym pozostają natomiast doskonale znane zalety postaci znormalizowanej, takie jak łatwa konserwacja, rozwój schematu oraz porządek, jaki wprowadza.

Bibliografia

1. Jajeśnica Ł., Piórkowski A.: WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH, STUDIA INFORMATICA 2010 Volume 31, Kraków 2010.
2. http://stareaneksy.pwn.pl/historia_ziemi/przyklady/?pokaz=tabela, ostatni dostęp 13.06.2024 godzina 15:00.