Edyta Skwarczek
Akademia Górniczo – Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej

SPRAWOZDANIE NA TEMAT WYDAJNOŚCI ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

1. Wprowadzenie

Celem tego dokumentu jest przeanalizowanie wydajnościowych aspektów normalizacji schematów baz danych poprzez porównanie dwóch podejść: schematu płatka śniegu (znormalizowanego) oraz schematu gwiazdy (zdemoralizowanego). Poniższe opracowanie bada konstrukcję dużych baz danych i hurtowni danych na przykładzie wymiaru czasu geologicznego, który tworzy hierarchię jednostek geochronologicznych. Dodatkowo, przeprowadzone eksperymenty sprawdzają wydajność złączeń i zagnieżdżeń skorelowanych w systemach zarządzania bazami danych Microsoft Sql Server Management (MSSMS) i PostgreSQL.

1.1. Normalizacja schematów tabel w bazach danych

Normalizacja to proces organizowania danych w bazie danych. Obejmuje ona tworzenie tabel i ustanawianie relacji między tymi tabelami zgodnie z regułami zaprojektowanymi zarówno w celu ochrony danych, jak i zwiększenia elastyczności bazy danych poprzez wyeliminowanie nadmiarowości i niespójnej zależności.

Istnieje kilka reguł normalizacji bazy danych. Każda reguła jest nazywana "formularzem normalnym"(FN). Jeśli zostanie zaobserwowana pierwsza reguła, mówi się, że baza danych ma "pierwszą normalną formę". Jeśli zostaną zaobserwowane pierwsze trzy reguły, baza danych jest uważana za "trzecią normalną formę". Chociaż możliwe są inne poziomy normalizacji, trzecia normalna forma jest uważana za najwyższy poziom niezbędny dla większości aplikacji.

• 1NF – Pierwsza postać normalna wymaga, aby każda krotka była atomowa, co oznacza, że wartości nie mogą być dalej podzielone. Relacje w modelu relacyjnym zawsze spełniają ten warunek.

• 2NF – Druga postać normalna wymaga, aby każdy atrybut wtórny był w pełni funkcyjnie zależny od całego klucza głównego, eliminując częściowe zależności od klucza.

• 3NF – Trzecia postać normalna wymaga, aby relacja była w drugiej postaci normalnej i aby każdy atrybut wtórny był bezpośrednio zależny wyłącznie od klucza głównego, eliminując zależności przejściowe.

1.2. Schematy hurtowni danych

Według Williama Inmona, hurtownia danych to zbiór danych wyróżniający się kilkoma kluczowymi cechami. Przede wszystkim jest uporządkowany tematycznie i zintegrowany, co oznacza, że dane są skonsolidowane z różnych źródeł i ułożone według określonych tematów. Hurtownia danych zawiera również wymiar czasowy, co pozwala na analizę danych w kontekście czasu. Jest to system nieulotny, co gwarantuje, że dane są trwałe i niezmienne po ich wprowadzeniu. Hurtownia danych wspomaga podejmowanie decyzji oraz

przetwarzanie informacji dla celów strategicznych i analitycznych, umożliwiając organizacjom lepsze zarządzanie i analizę swoich danych.

Schemat gwiazdy

Schemat gwiazdy cechuje się prostą strukturą i wysoką efektywnością zapytań dzięki małej liczbie powiązań między tabelami. Jednak ładowanie danych do tabel wymiarów jest czasochłonne z powodu denormalizacji i redundancji danych.

Tabela faktów zawiera miary (numeryczne wartości opisujące fakty) i klucze obce do tabel wymiarów, które mają wartości opisowe. Klucz główny tabeli faktów składa się ze wszystkich jej kluczy obcych. Tabela faktów może zawierać informacje zarówno na poziomie detalicznym, jak i zagregowanym.

Schemat płatka śniegu

Schemat płatka śniegu różni się od schematu gwiazdy kilkoma cechami. Pomimo łatwiejszej struktury do modyfikacji, wydajność zapytań jest niższa z powodu większej liczby relacji, co sprawia, że zapytania są wolniejsze. Jednak czas ładowania danych do tabel jest krótszy, ponieważ normalizacja zmniejsza redundancję.

2. Tabela geochronologiczna

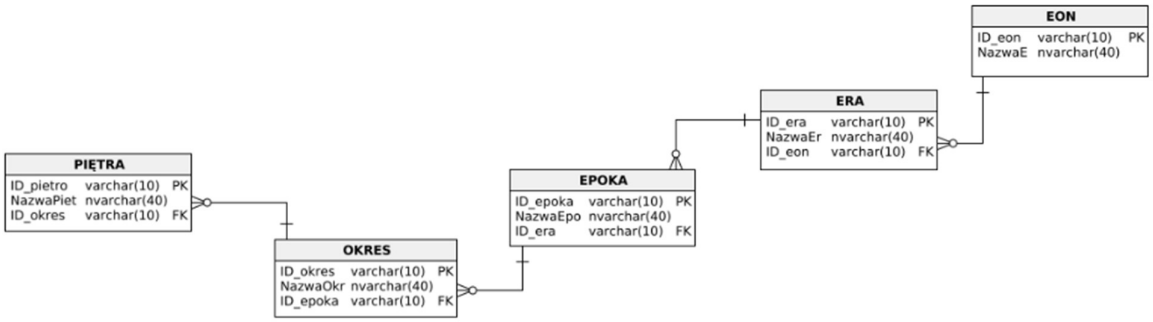
Tabela geochronologiczna jest schematem obrazującym przebieg historii Ziemi na podstawie następstwa procesów geologicznych i układu warstw skalnych.

Wiek (mln lat)	Eon	Era	Okres	Epoka
0,010	**FANEROZOIK**	**Kenozoik**	**Czwartorzęd**	Holocen
1,8				Plejstocen
22,5			**Neogen**	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

3. Konstrukcja wymiaru geochronologicznego

W niniejszym opracowaniu skupiono się na konstrukcji tabeli geochronologicznej w dwóch przypadkach:

- schemacie znormalizowanym (płatka śniegu, rys. 1);



Rys. 1. Znormalizowany schemat tabeli geochronologicznej, wygenerowany za pomocą Vertabelo Database Modeler

- schemacie zdenormalizowanym (schemat gwiazdy).

Formę zdenormalizowaną tabeli geochronologicznej osiągnięto tworząc jedną tabelę Tabela Straty (rys. 2), zawierającą wszystkie dane z powyższych tabel.

Rys. 2. Zdenormalizowany schemat tabeli geochronologicznej, wygenerowany za pomocą Vertabelo Database Modeler

Dokonano tego za pomocą złączenia naturalnego, obejmującego wszystkie tabele tworzące hierarchię.

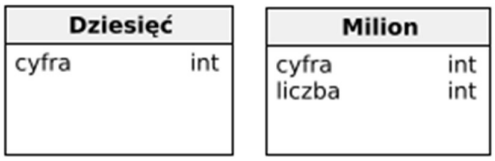
4. Testy wydajności

W testach skupiono się na porównaniu wydajności złączeń oraz zapytań zagnieżdżonych, wykonywanych na tabelach o dużej liczbie danych. Testy wykonano na dwóch różnych maszynach identycznych pod względem konfiguracji sprzętowej oraz oprogramowania. Przetestowano najpopularniejsze darmowe rozwiązania bazodanowe:

- MSSMS,
- PostgreSQL.

W zapytaniach testowych łączono dane z tabeli geochronologicznej z syntetycznymi danymi o rozkładzie jednostajnym z tabeli *Milion*, wypełnionej kolejnymi liczbami naturalnymi od 0 do 999 999. Tabela *Milion* została utworzona na podstawie odpowiedniego autozłączenia tabeli *Dziesięć* wypełnionej liczbami od 0 do 9

Schematy tabel przedstawiono na rysunku 3.



Rys. 3. Schemat tabel Dziesięć i Milion

4.1. Konfiguracja sprzętowa i programowa

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

- CPU: Intel® Core™ i5-1035G1 (4 rdzenie, 8 wątków, 1.00-3.60 GHz, 6MB cache)
- RAM: 8 GB (DDR4, 3200 MHz)
- SSD: 1 x M.2, PCIe, 512 GB
- S.O.: Windows 10 Home

Jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

- Microsoft SQL Server Management Studio - 19.3
- PostgreSQL, wersja 15.7-2

Testy wykonywano wielokrotnie dla każdego systemu zarządzania bazą danych.

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:

Pierwszy etap obejmował zapytania bez nałożonych indeksów na kolumny danych (jedynymi indeksowanymi danymi były dane w kolumnach będących kluczami głównymi poszczególnych tabel,),

- w drugim etapie nałożono indeksy na wszystkie kolumny biorące udział w złączeniu.

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
    liczby.milion m
INNER JOIN
    tabela_stratygraficzna.TabelaStraty t
```

```
ON
    m.liczba % 95 = CAST(RIGHT(t.ID_pietra, LEN(t.ID_pietra) - 3) AS INT);
```

- 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
    liczby.milion m
INNER JOIN
    tabela_stratygraficzna.Pietra p
ON
    m.liczba % 95 = CAST(RIGHT(p.ID_pietro, LEN(p.ID_pietro) - 3) AS INT)
INNER JOIN
    tabela_stratygraficzna.Epoka ep
ON
    p.ID_epoka = ep.ID_epoka
INNER JOIN
    tabela_stratygraficzna.Okres o
ON
    ep.ID_okres = o.ID_okres
INNER JOIN
    tabela_stratygraficzna.Era er
ON
    o.ID_era = er.ID_era
INNER JOIN
    tabela_stratygraficzna.Eon eo
ON
    er.ID_eon = eo.ID_eon;
```

- 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
    liczby.milion m
WHERE
    m.liczba % 95 =
    (SELECT
        CAST(RIGHT(p.ID_pietro, LEN(p.ID_pietro) - 3) AS INT)
    FROM
        tabela_stratygraficzna.Pietra p
    WHERE
        m.liczba % 95 = CAST(RIGHT(p.ID_pietro, LEN(p.ID_pietro) - 3) AS INT));
```

- 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
    liczby.milion m
```

```
WHERE
    m.liczba % 95 IN
    (SELECT
        CAST(RIGHT(p.ID_pietro, LEN(p.ID_pietro) - 3) AS INT)
    FROM
        tabela_stratygraficzna.Pietra p
    INNER JOIN
        tabela_stratygraficzna.Epoka ep
    ON
        p.ID_epoka = ep.ID_epoka
    INNER JOIN
        tabela_stratygraficzna.Okres o
    ON
        ep.ID_okres = o.ID_okres
    INNER JOIN
        tabela_stratygraficzna.Era er
    ON
        o.ID_era = er.ID_era
    INNER JOIN
        tabela_stratygraficzna.Eon eo
    ON
        er.ID_eon = eo.ID_eon);
```

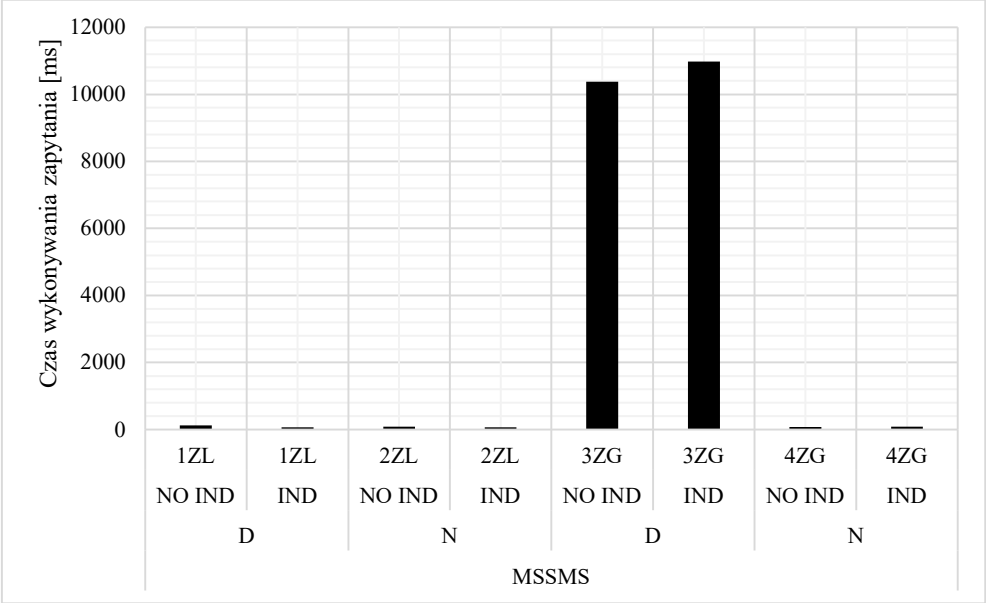
5. Wyniki testów

Każdy test przeprowadzono wielokrotnie, wyniki skrajne pominięto.

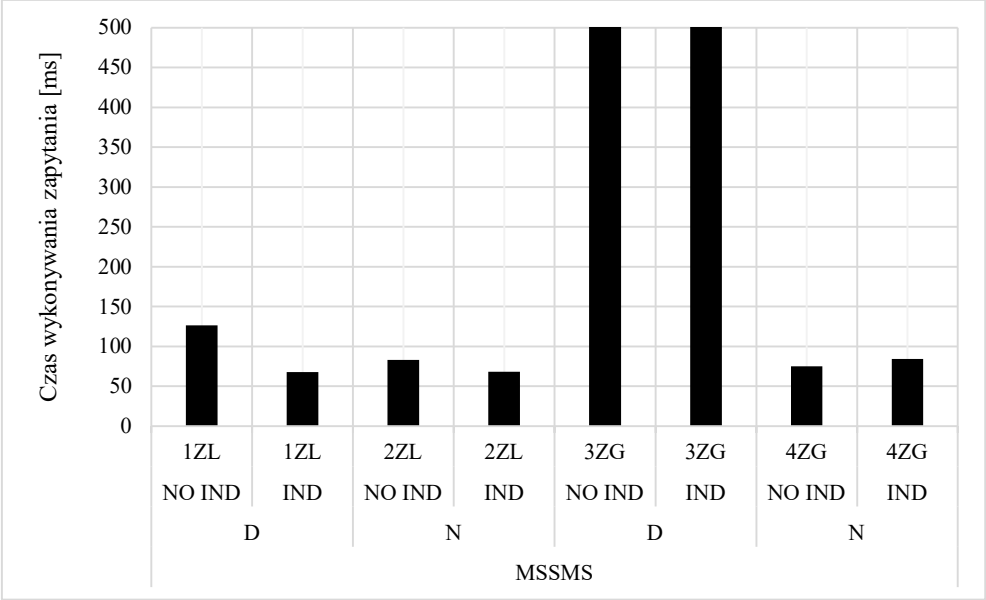
Czasy wykonania zapytań 1 ZL, 2 ZL, 3 ZG i 4 ZG [ms]

	1ZL			**2ZL**			**3ZG**			**4ZG**		
BEZ INDEKSÓW	MIN	ŚR	MAX	MIN	ŚR	MAX	MIN	ŚR	MAX	MIN	ŚR	MAX
MSSMS	64,88	126,3	2621	71,20	82,77	1703	9124	10374	49575	70,77	74,83	724
PostgreSQL	2085	2742,6	3584	1601	2192,6	2912	48525	49150	49614	639	787	1158
Z INDEKSAMI												
MSSMS	60,91	67,9	914	59,09	68,22	670	9712	10970	48508	71,77	84,31	568
PostgreSQL	570	716,2	914	532	583,4	670	46479	48075,6	49482	568	633,8	696

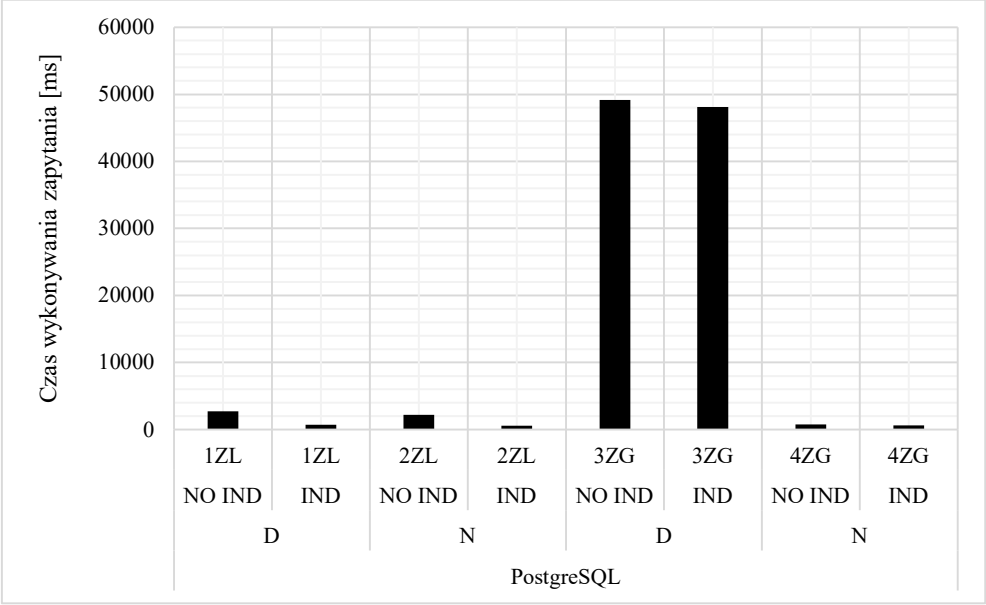
Analizę wyników ułatwiają wykresy (rys. 4,5 dla MSSMS, 6 i 7, dla PostgreSQL, oraz 8 i 9 dla obu systemów) – ze względu na dość duże wartości pojedynczych przypadków rozważono dwie wersje – pełna skala liniowa i skala liniowa częściowa (aby ułatwić porównanie niskich wartości). Wyniki zestawiono wysuwając na pierwszy plan związki z tezą artykułu na którym bazuje sprawozdanie – czy wersja znormalizowana (N) jest wolniejsza czy szybsza od wersji zdenormalizowanej (D).



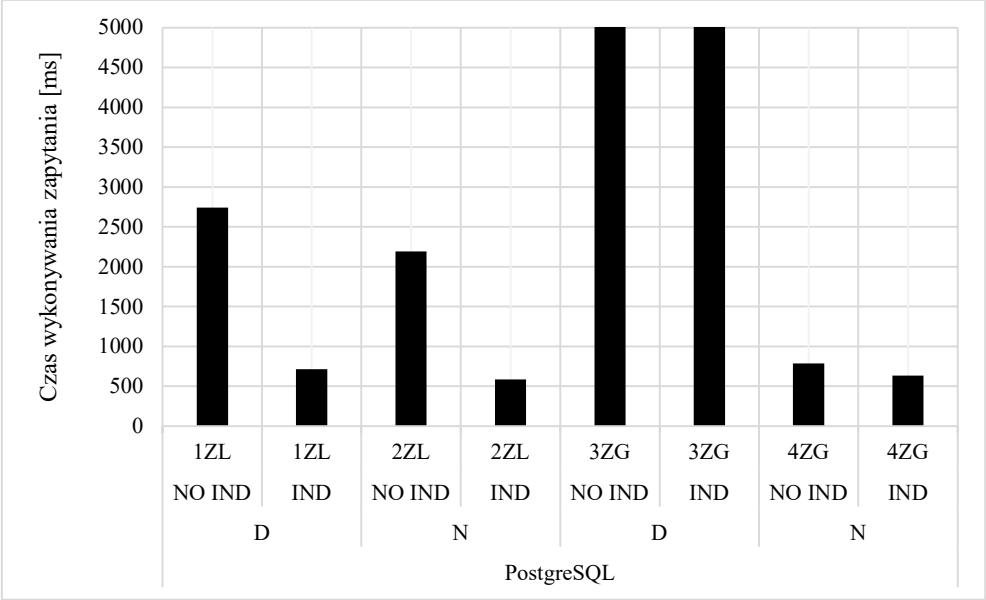
Rys. 4. Wyniki testów w ujęciu celu normalizacji w systemie MSSMS



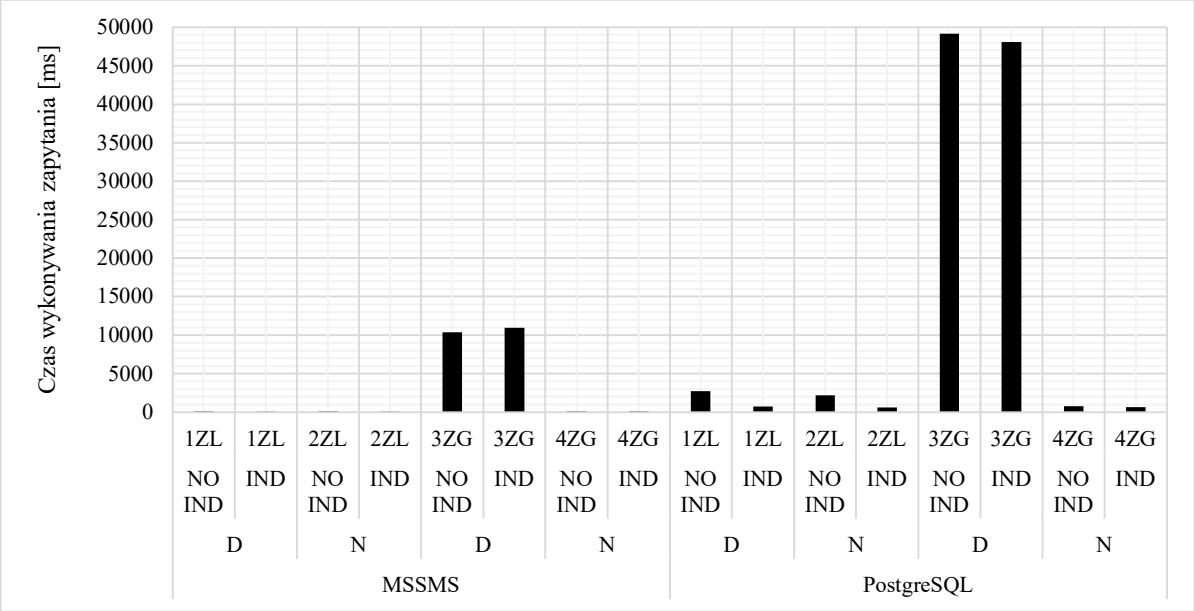
Rys. 5. Wyniki testów w ujęciu celu normalizacji dla niższych wartości w systemie MSSMS



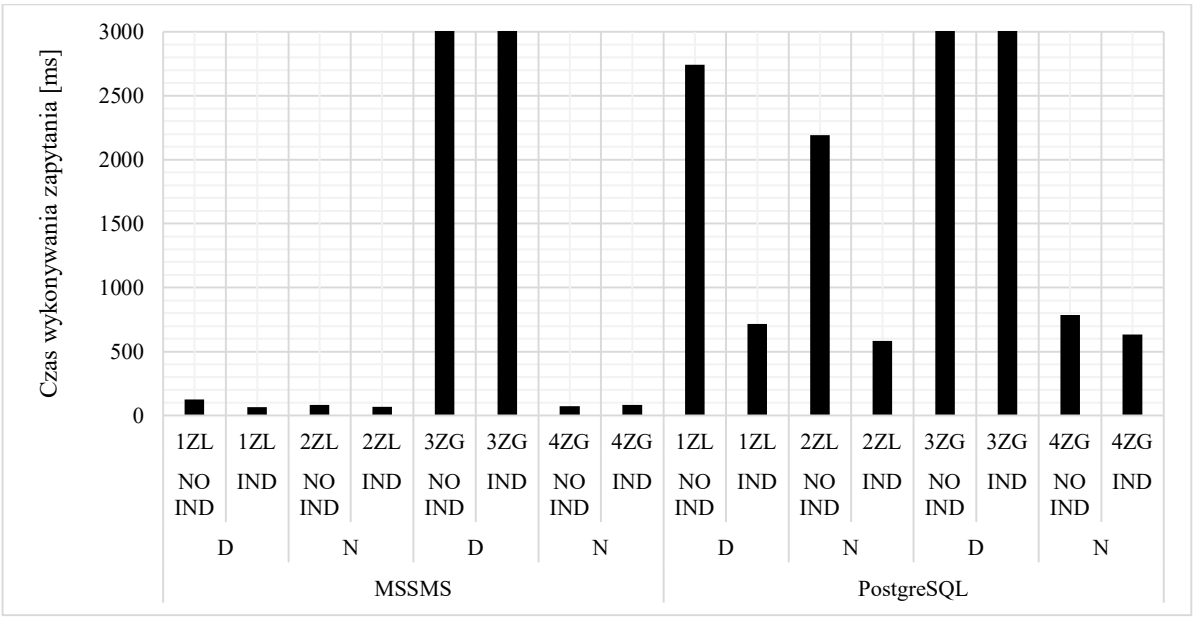
Rys. 6. Wyniki testów w ujęciu celu normalizacji w systemie PostgreSQL



Rys. 7. Wyniki testów w ujęciu celu normalizacji dla niższych wartości w systemie PostgreSQL



Rys. 8. Wyniki testów w ujęciu celu normalizacji w obu testowanych systemach zarządzania bazami danych



Rys. 9. Wyniki testów w ujęciu celu normalizacji dla niższych wartości w obu testowanych systemach zarządzania bazami danych

6. Wnioski

Dla zapytań 1 i 2, indeksowanie przyniosło zadowalające wyniki, przyspieszając czasy wykonywania obliczeń nawet o prawie 74% w przypadku PostgreSQL. Jednak dla testów z zagnieżdżeniami wyniki nie były już tak dobre, a nawet nieznacznie gorsze po indeksacji w przypadku MSSMS. W związku z tym nie można jednoznacznie stwierdzić, że indeksowanie zawsze przyspiesza szybkość obliczeń.

W analizie porównawczej dwóch testowanych systemów wykazano, że MSSMS sprawdził się znacznie lepiej niż PostgreSQL, osiągając nawet 755% lepsze wyniki zarówno przed, jak i po indeksacji. Wyjątkiem był trzeci test, w którym MSSMS okazał się prawie czterokrotnie gorszy od PostgreSQL w obu przypadkach.

Zauważono również, że normalizacja w większości przypadków prowadzi do zwiększenia wydajności. Ma ona swoje zalety, takie jak lepsza organizacja danych w porównaniu do tabel nieznormalizowanych. Dodatkowo, tabele znormalizowane są bardziej elastyczne pod względem dalszego rozwoju, co ułatwia ich modyfikowanie.

BIBLIOGRAFIA

1. „Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych” Łukasz JAJEŚNICA, Adam PIÓRKOWSKI, 30 stycznia 2010 r.
2. „Opis podstaw normalizacji bazy danych” Microsoft Learn, 14 lipca 2023 r.
3. „Hurtowanie Danych” Mariusz Żynel, Uniwersytet w Białymstoku, 31 stycznia 2017 r.