JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 28 (2023) 296–303
Received: 19 June 2023
Accepted: 29 June 2023

The comparative performance analysis of selected relational database systems

Analiza porównawcza wydajności wybranych relacyjnych systemów baz danych

Szymon Schab*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
The objective of this study was to carry out a performance analysis of the following database systems: MySQL, PostgreSQL and Microsoft SQL Server. For this purpose scripts were used to measure execution times of selecting, updating and inserting data. Furthermore, three data sets were utilized consisting of 100, 1 000 and 10 000 rows. The experiment included nine cases depending on the query type and the data set. For each case, thirty five test trials were conducted while first five trials were ignored i.a. because of cache storage. The statistical test was performed for the results and the trials in which the DBMS achieved best times were counted. For each case best systems were acknowledged and the most efficient system of the experiment was determined along with systems for each operation type.

Keywords: performance analysis; MySQL; PostgreSQL; Microsoft SQL Server

Streszczenie
Celem pracy było przeprowadzenie analizy wydajnościowej trzech relacyjnych systemów zarządzania bazami danych: MySQL, PostgreSQL i Microsoft SQL Server. W badaniu wykorzystano skrypty mierzące długości czasów operacji wstawiania, aktualizacji i zwracania danych, a także trzy zestawy danych liczące kolejno 100, 1 000 i 10 000 rekordów. Eksperyment składał się z dziewięciu przypadków uwzględniających rodzaj operacji i wariant zestawu danych, dla których wykonano po trzydzieści pięć prób, przy czym pierwsze pięć prób pominięto m.in. ze względu na kwestie przechowywania danych w pamięci podręcznej. Otrzymane wyniki sprawdzono pod kątem istotności różnic, a następnie dla każdego z przypadków zliczono liczbę prób, w których oprogramowania uzyskały najlepsze wyniki. Na końcu wskazano i policzono najlepsze systemy dla poszczególnych przypadków i wyznaczono najwydajniejszy system dla całego badania oraz systemy dla testowanych rodzajów operacji.

Słowa kluczowe: analiza wydajności; MySQL; PostgreSQL; Microsoft SQL Server

*Corresponding author
Email address: **szymon.schab@pollub.edu.pl** (S. Schab)

1. Wstęp

Współczesne systemy bazodanowe są podstawą wielu systemów informatycznych. Rynek oferuje mnóstwo narzędzi pozwalających na zarządzanie systemami bazodanowymi, spośród których jednymi z najpopularniejszych stały się relacyjne systemy baz danych. Rosnące zapotrzebowanie na takie programy spowodowało, że powstało ich wiele proponując tym samym zróżnicowane poziomy wydajności pracy. W pracach [1-13] skupiona została uwaga na efektywności systemów DBMS uzyskanej w testach wykorzystujących m.in. zapytania SQL. Otrzymane wyniki pozwoliły stwierdzić, które oprogramowanie i w jakich scenariuszach sprawdziło się najlepiej.

Zdolność platformy do przystosowania się do zwiększonych wymagań przetwarzania danych odgrywa kluczową rolę w podejmowaniu decyzji, dlatego celem pracy [14] było zbadanie wydajności dwóch silników relacyjnych baz danych - MySQL InnoDB i Microsoft SQL Server 2012 dla środowisk DSS. Przeprowadzone eksperymenty pozwoliły ocenić osiągi i przydatność badanych silników do małych i średniej wielkości środowisk wspomagania decyzji (DSS). W testach użyto programu Star Schema Benchmark, który wykorzystywał przebiegi zapytań (ang. query flight), z których każdy składał się z zapytań o różnej selektywności. Query Flight Q1 wykonywał łączenie *join* między tabelą Lineorder a tabelą Date Dimension, Query Flight Q2 łączył tabele Lineorder, Date, Part i Supplier, a także agregował oraz sortował dane według roku i marki, Query Flight Q3 zwracał całkowity przychód dla transakcji Lineorder w danym regionie w określonym czasie, Query Flight Q4 łączył wszystkie tabele i zwracał łączny zysk pogrupowany według roku i nacji. W badaniu użyty został program Star Schema Benchmark w celu utworzenia zestawów danych o rozmiarach 1 GB, 3 GB, 6 GB, 12 GB i 24 GB. Przy użyciu programu DBgen wygenerowano tabele z 6, 18, 36, 64, 71, 144 milionami rekordów. Następnie przetestowano zapytania SSB dla różnych zestawów danych, każdy z 3 uruchomieniami i obliczono średni czas wykonania zapytania. Dla zestawów danych do 12 GB, MySQL InnoDB uzyskał czasy zapytań, które można uznać za akceptowalne, ale w przypadku eksploracji i analizy większych zbiorów danych pomiędzy tymi dwoma DBMS, lepszym silnikiem okazał się Microsoft SQL Server 2012.

2. Testowane technologie

MySQL jest darmowym systemem zarządzania relacyjnymi bazami danych wydanym w 1995 roku przez szwedzką firmę MySQL AB. System ten pomaga m.in. stworzyć i wdrożyć w projekcie relacyjną bazę danych, wygenerować jej kopie zapasowe czy sprawdzić integralność bazy. Omawiany program bazuje na koncepcji „open-source" pozwalającą każdemu na wgląd oraz ingerencję w kod źródłowy oprogramowania. Sam MySQL korzysta z języka SQL (ang. Structured Query Language), a dane zarządzane są w sposób zgodny z założeniami relacyjnego modelu baz danych.

Microsoft SQL Server to system zarządzania bazami danych wydany w 1989 roku przez firmę Microsoft. System ten jest stosowany na szeroką skalę przez przedsiębiorstwa różnego rodzaju, w tym przez wiele firm informatycznych. Oparty jest na języku SQL, choć sam implementuje Transact-SQL. MS SQL Server można uruchomić na następujących platformach: Windows, Linux i Docker.

PostgreSQL to system zarządzania bazami danych wydany w 1989 przez grupę specjalistów na Uniwersytecie Kalifornijskim w Berkeley. Został on stworzony na takie platformy jak Unix, Linux, Windows oraz środowiska chmurowe np. Microsoft Azure i Amazon Web Services. Uważany jest za jeden z najbardziej rozwiniętych systemów na rynku. Oprogramowanie to bazuje na języku SQL, ale także obsługuje język procedur składowanych PL/pgSQL. PostgreSQL jest programem utworzonym zgodnie z koncepcją „open-source" zapewniającą otwarty dostęp do kodu źródłowego.

3. Cel i zakres pracy

Celem pracy było przeprowadzenie analizy wydajnościowej następujących relacyjnych systemów zarządzania bazami danych: MySQL, PostgreSQL i Microsoft SQL Server. Badanie uwzględniało serię testów wykorzystujących skrypty mierzące wartości czasów poszczególnych rodzajów operacji. Weryfikowane typy zapytań to: wstawianie rekordu (INSERT), aktualizacja rekordu (UPDATE) i zwrócenie konkretnych wierszy (SELECT). Eksperyment przeprowadzono dla każdego systemu baz danych. W celu otrzymania kompleksowej i wiarygodnej oceny wydajności zastosowano zestawy danych zawierające odpowiednio 100, 1 000 i 10 000 rekordów. Każdy z tych zestawów uwzględniono we wszystkich rodzajach operacji.

Teza pracy zakłada, że system bazodanowy Microsoft SQL Server jest najwydajniejszy ze wszystkich testowanych systemów. W pracy sformułowano także pytania badawcze mające za zadanie uzyskać konkretną wiedzę w ramach określonych zagadnień. Pytania te prezentowały się następująco:

1. Który z testowanych systemów baz danych jest najwydajniejszy w przypadku zapytania zwracającego rekordy?
2. Który z testowanych systemów baz danych jest najwydajniejszy w przypadku zapytania aktualizującego rekordy?
3. Który z testowanych systemów baz danych jest najwydajniejszy w przypadku zapytania wstawiającego rekordy?

4. Plan badań

Plan eksperymentu badawczego składał się z następujących etapów:

1. Przygotowanie środowiska testowego:
   a) Zainstalowanie i skonfigurowanie systemów zarządzania bazami danych MySQL, PostgreSQL i Microsoft SQL Server.
   b) Wygenerowanie trzech baz danych wraz z tabelami o jednakowej strukturze.
   c) Zaimportowanie danych testowych do tabel w bazach danych.
   d) Przygotowanie skryptów wykonujących operacje wstawiania (insert), aktualizacji (update) i zwracania rekordów (select) dla każdego z systemów bazodanowych.
2. Przeprowadzenie badań:
   a) Wykonanie operacji wstawiania danych i zmierzenie czasów dla trzech grup rekordów liczących odpowiednio 100, 1 000 i 10 000 wierszy.
   b) Wykonanie operacji aktualizacji danych i zmierzenie czasów dla trzech grup rekordów liczących odpowiednio 100, 1 000 i 10 000 wierszy.
   c) Wykonanie operacji zwracania danych i zmierzenie czasów dla trzech grup rekordów liczących odpowiednio 100, 1 000 i 10 000 wierszy. Operacja *select r*ealizowana jest z uwzględnieniem projekcji oraz wyszukiwania danych z użyciem klauzuli *where* i podzapytania.
3. Analiza wyników:
   a) Przeprowadzenie testu statystycznego dla zestawów wyników.
   b) Wskazanie i zliczenie zwycięzców dla poszczególnych przypadków oraz wyznaczenie zwycięzcy dla całości eksperymentu i testowanych rodzajów operacji.
   c) Wyciągnięcie wniosków dotyczących wydajności testowanych systemów w kontekście przeprowadzonych operacji.

Dla wszystkich systemów bazodanowych ustalono dziewięć przypadków testowych uwzględniających rodzaj operacji i wariant zestawu danych. Dla każdego przypadku zrealizowano trzydzieści pięć prób, przy czym pierwsze pięć prób zostało pominiętych ze względu na kwestie optymalizacji bazy danych i kwestie przechowywania danych w pamięci podręcznej. W dalszej kolejności zestawiono ze sobą rezultaty otrzymane przez systemy dla poszczególnych przypadków i wykorzystano odpowiedni test statystyczny do zbadania istotności różnic zestawów wyników. Następnie dla każdego z przypadków zliczono liczbę prób, w których oprogramowania uzyskały najlepsze wyniki oraz wskazano i policzono zwycięzców dla tych przypadków.

Przygotowanie systemów bazodanowych odbyło się na komputerze o specyfikacji przedstawionej w Tabeli 1. Konfigurację serwerów przedstawiono w Tabeli 2.

Znajdują się tutaj takie parametry jak rozmiar bufora czy maksymalna liczba połączeń. Ten pierwszy wskazuje na rozmiar w pamięci zarezerwowanej w celu zapisywania w pamięci podręcznej danych tabel oraz indeksów i ich odczytu. W przypadku MS SQL Server wspomniany rozmiar jest dobierany przez system bazodanowy. Maksymalna liczba połączeń odpowiada za największą liczbę jednoczesnych połączeń na serwerze.

Tabela 1: Specyfikacja komputera do badań

rodzaj podzespołu	podzespół
procesor	Intel Core i7-4790k, 4,00 GHz, 8 MB, 4 rdzenie, architektura 64-bitowa
pamięć RAM	Mushkin Stealth, 8 GB, 1 600 MHz, CL 9, DDR3
dysk	Samsung 870 EVO, 500 GB, SSD
system operacyjny	Windows 10 Pro, wersja 64-bitowa

Tabela 2: Konfiguracja serwerów

	MySQL	MS SQL Server	PostgreSQL
nazwa i wersja serwera	MySQL Community Server – GPL 8.0.32	SQL Server 2022 (RTM) - 16.0.1000.6	PostgreSQL 15.3
port	3306	1434	5432
adresy	wszystkie	127.0.0.1	wszystkie
rozmiar bufora	128 MB	-	128 MB
maksymalna liczba połączeń	151	32 767	100

4. Środowisko testowe

4.1. Baza danych

Na wszystkich testowanych systemach baza danych opierała się o ten sam schemat przedstawiający model dla sklepu internetowego zajmującego się technologiami konsumenckimi. Struktura bazy zawierała trzynaście tabel, a każda z tabel skupiała się na konkretnym aspekcie dotyczącym wspomnianego biznesu, np. tabela „customer_credit_debit_card" zbierała informacje na temat zapisanych przez użytkownika kartach płatniczych (kredytowych i debetowych), a tabela „customers" na temat danych klientów. Jeśli chodzi o pozostałe obiekty to tabela „orders" zawierała dane o zamówieniach, „order_items" o elementach zamówienia, „products" o produktach, „customer_address" o zapisanych przez użytkownika adresach, „invoices" o fakturach, „receipts" o paragonach, „shipments" o wysyłkach, „shipment_items" o elementach wysyłki, „product_categories" o kategoriach produktów, „employees" o pracownikach, a „payments" o płatnościach. W tabelach uwzględniono takie indeksy jak klucz podstawowy, klucz obcy i wartość unikalna. Dane zaimportowane do bazy danych zostały wygenerowane przy użyciu internetowego narzędzia Mockaroo [15]. Program ten pozwolił na utworzenie rekordów dla tabel zgodnie z obowiązującym schematem. Przy generowaniu danych zostały uwzględnione takie aspekty jak możliwość wstawienia wartości NULL do kolumny oraz typy danych.

4.2. Skrypty

Struktura kodu skryptów była bardzo podobna dla wszystkich trzech systemów, a programy zwracały ten sam rodzaj informacji, tj. tabelę z numerami prób i wartościami czasów dla wskazanego zestawu danych i typu polecenia.

Program opierał się na dwóch pętlach: zewnętrznej kontrolującej liczbę prób i wewnętrznej kontrolującej liczbę wykonań polecenia. Pomiary przeprowadzono w taki sposób, że zmienna „@start_" została umieszczona tuż przed fragmentem kodu testowanej operacji, a zmienna „@end_" zaraz po nim. Obu tym zmiennym przyporządkowano wartości aktualnego czasu mierzonego w sekundach z dokładnością do sześciu miejsc po przecinku. Potem obliczono różnicę i przypisano ją zmiennej „@singleQueryDuration", której wartość dodano do zmiennej „@totalDuration". Po wyjściu z pętli wewnętrznej następowało wprowadzenie uzyskanego rezultatu do tymczasowej tabeli i wyzerowanie zmiennych. Po wykonaniu wszystkich prób program wyświetlił tabelę z wynikami. Co do poleceń, instrukcja SELECT zwracała tylko jeden rekord z informacjami o kliencie, którego zamówienia:

- były o wartości przynajmniej 5 000 USD,
- zostały już dostarczone,
- zostały opłacone przy użyciu karty kredytowej lub debetowej,
- mają paragon z datą wynoszącą najwcześniej 2023-01-15,
- składały się tylko z nadal dostępnych produktów.

W przypadku polecenia aktualizującego rekordy (Listing 1) jedyną zmianą w skrypcie było zastąpienie fragmentu z poleceniem „SELECT" kodem z instrukcją „UPDATE". Instrukcja ta zmniejszała ceny i zwiększała liczbę produktów należących do tej samej kategorii. Podobnie wyglądało przygotowanie skryptu wstawiającego dane. Po zamianie odpowiednich instrukcji program poprawnie zmierzył długości czasów dla polecenia „INSERT", które wprowadzało do tabeli „order_items" jeden rekord z informacjami dla konkretnego zamówienia.

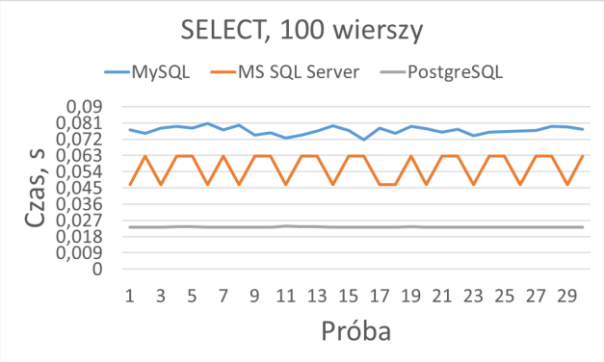
5. Rezultaty

5.1. Zwracanie danych

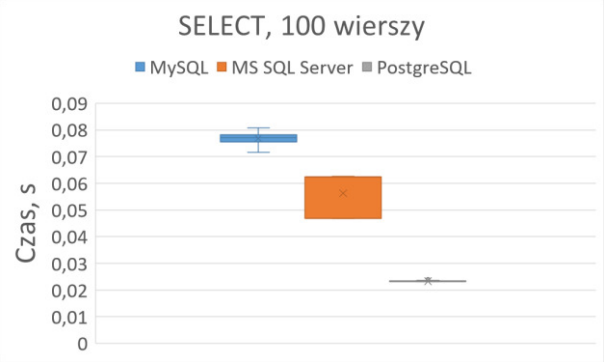
Na Rysunku 1 znajduje się wykres czasów uzyskanych przez trzy systemy bazodanowe dla pierwszego przypadku, czyli dla operacji zwracania 100 rekordów. Tutaj zwycięzcą okazał się PostgreSQL osiągając znacznie mniejsze wartości od konkurencji. Różnica między MS SQL Server wynosiła około dwukrotności długości czasów PostgreSQL, a dla MySQL ponad trzykrotność długości czasów. Zwycięskiemu systemowi udało się utrzymać najlepszą stabilność w swoich wynikach (Rysunek 2). Świadczy o tym nie tylko brak wartości odstających, ale również najwęższa część pudełkowa wykresu, najbliższe położenie mediany i kwartyli, a także najmniejsza różnica pomiędzy wartością maksymalną i minimalną.

Listing 1: Fragment kodu programu mierzącego czasy operacji aktualizacji danych dla systemu MS SQL Server

```
SET @start_ = DATEDIFF_BIG(MICROSECOND,'1970-01-01 00:00:00', SYSDATETIME()) / CAST(1000000 AS DECIMAL(30,6))
UPDATE products
    SET products.product_price = products.product_price * 0.99,
        products.product_quantity_in_stock = products.product_quantity_in_stock + 5
    FROM products
    INNER JOIN product_categories ON
    products.product_categories_id_product_categories = product_categories.id_product_categories
    WHERE product_categories.id_product_categories = 3
SET @end_ = DATEDIFF_BIG(MICROSECOND,'1970-01-01 00:00:00', SYSDATETIME()) / CAST(1000000 AS DECIMAL(30,6))
SET @singleQueryDuration = @end_ - @start_
SET @totalDuration = @totalDuration + @singleQueryDuration
```



Rysunek 1: Wykres liniowy dla operacji zwracania 100 rekordów.

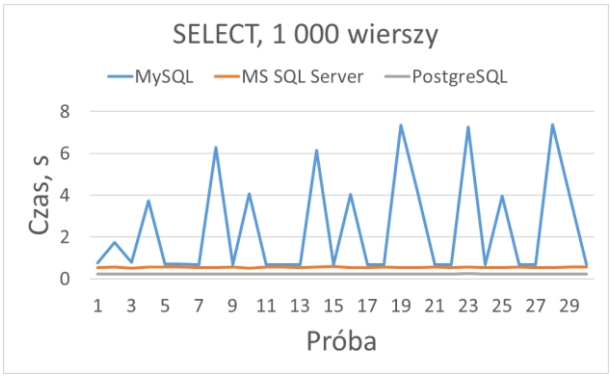


Rysunek 2: Wykres pudełkowy dla operacji zwracania 100 rekordów.

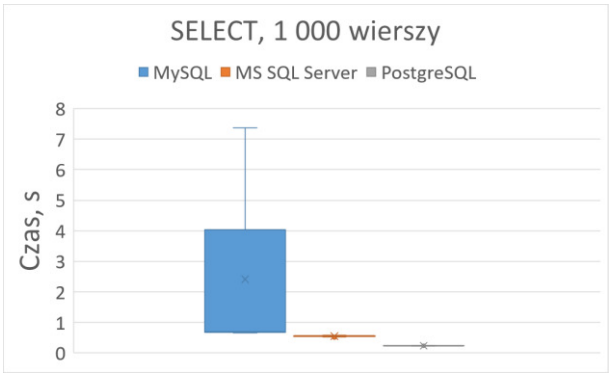
Na Rysunku 3 przedstawiono wykres czasów dla drugiego przypadku - operacji zwracania 1 000 rekordów. Ponownie najlepsze rezultaty otrzymał PostgreSQL. Wartości w tym przypadku były przynajmniej dwukrotnie mniejsze od czasów dla systemu MS SQL Server, a także systemu MySQL. Warto zauważyć poziom stabilności wyników dla PostgreSQL i MS SQL Server (Rysunek 4), które prezentują się bardzo dobrze w porównaniu z ogromną niestabilnością w rezultatach uzyskanych przez MySQL.

Rysunek 5 ilustruje wykres długości czasów trwania prób dla trzeciego przypadku, czyli dla operacji zwracania 10 000 rekordów. PostgreSQL znów zajął pierwsze miejsce, podczas gdy MS SQL Server zajął drugie, a MySQL trzecie. Różnice między rezultatami systemu PostgreSQL a MS SQL Server są ponad dwukrotne, zaś dla MySQL'a są one ponad ośmiokrotne. Stabilność wyników (Rysunek 6) prezentuje się najlepiej dla PostgreSQL i MS SQL Server, a najgorzej dla MySQL.

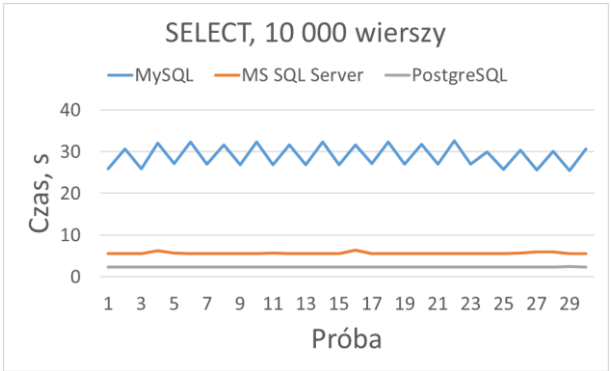
W trzech pierwszych przypadkach badania zaobserwowano, że MySQL zwykle osiągał najgorsze wartości, podczas gdy MS SQL Server znajdował się w środku rankingu, a PostgreSQL zostawał liderem. Wspomnianą zależność zauważono także w temacie stabilności systemów. Najstabilniejsze wartości osiągał PostgreSQL, a najmniej stabilne MySQL.



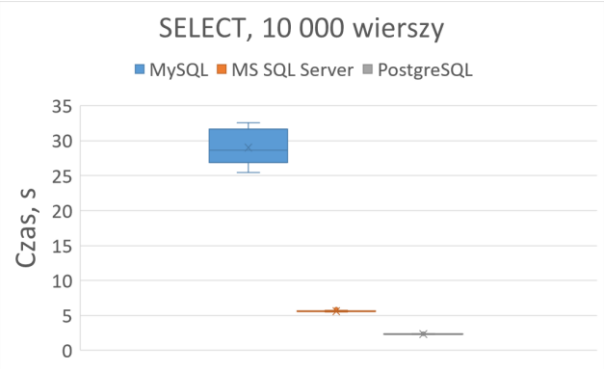
Rysunek 3: Wykres liniowy dla operacji zwracania 1 000 rekordów.



Rysunek 4: Wykres pudełkowy dla operacji zwracania 1 000 rekordów.



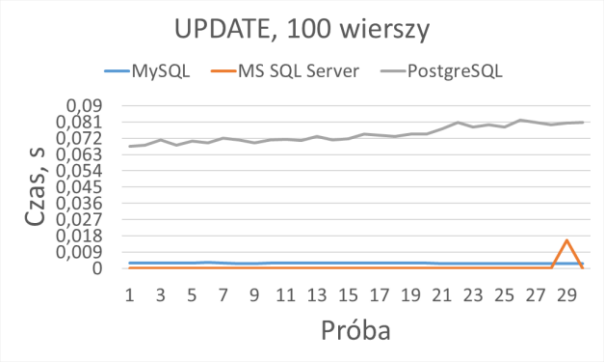
Rysunek 5: Wykres liniowy dla operacji zwracania 10 000 rekordów.



Rysunek 6: Wykres pudełkowy dla operacji zwracania 10 000 rekordów.

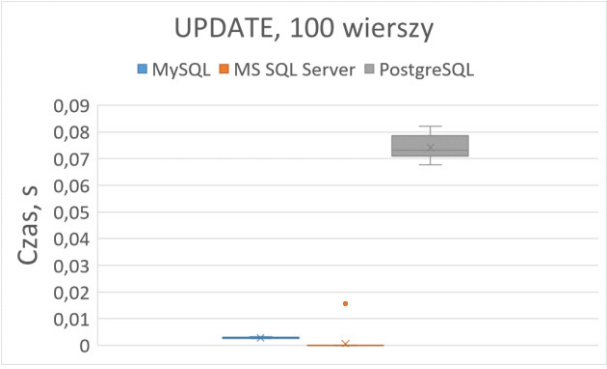
5.2. Aktualizacja danych

Aktualizacja danych to kolejny rodzaj operacji testowany na bazach. Dla przypadku ze stoma wierszami (Rysunek 7) zwycięzcą okazał się system MS SQL Server, który dla ustalonej precyzji do sześciu miejsc po przecinku uzyskiwał wyniki równe 0. Na drugim miejscu znalazł się MySQL, który swoimi rezultatami pokazał sprawność dla tego typu polecenia. Najgorszymi długościami czasów charakteryzował się PostgreSQL. Co ciekawe, czasy dla tego systemu rosły wraz z kolejnymi próbami. Działo się tak, dlatego, że PostgreSQL, przy aktualizacji danych, tworzy nowy wiersz i usuwa stary. Niestety, przy wielokrotnym aktualizowaniu dużej liczby rekordów oryginalne rekordy są nadal przechowywane w bazie, co powoduje zwiększenie jej rozmiaru, a tym samym wydłużenie wykonania zapytania. Różnice między wartościami MS SQL Server i MySQL wynosiły około 0,0028 s, a między MS SQL Server a PostgreSQL około 0,074 s. Największą stabilnością wyników odznaczyły się systemy MS SQL Server i MySQL, a najmniejszą PostgreSQL (Rysunek 8).

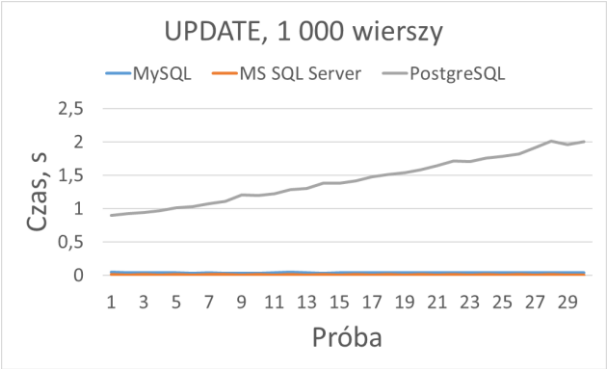


Rysunek 7: Wykres liniowy dla operacji aktualizacji 100 rekordów.

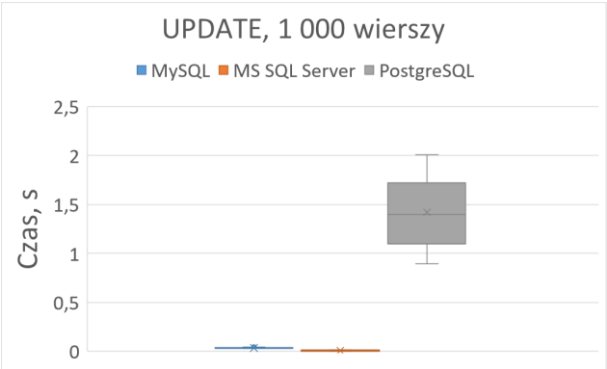
Ranking dla przypadku z 1 000 wierszy wyglądał podobnie jak poprzednio (Rysunek 9). Najkrótszymi czasami wykazał się MS SQL Server, zaraz po nim uplasował się MySQL, a na końcu znalazł się PostgreSQL. MS SQL Server uzyskał część wyników równą zero, a różnice w wartościach między tym systemem a MySQL były na poziomie około 0,028 s, podczas gdy dla PostgreSQL były one równo około 1,42 s. W kwestii stabilności najlepszymi rezultatami odznaczyły się MS SQL Server i MySQL (Rysunek 10).



Rysunek 8: Wykres pudełkowy dla operacji aktualizacji 100 rekordów.



Rysunek 9: Wykres liniowy dla operacji aktualizacji 1 000 rekordów.

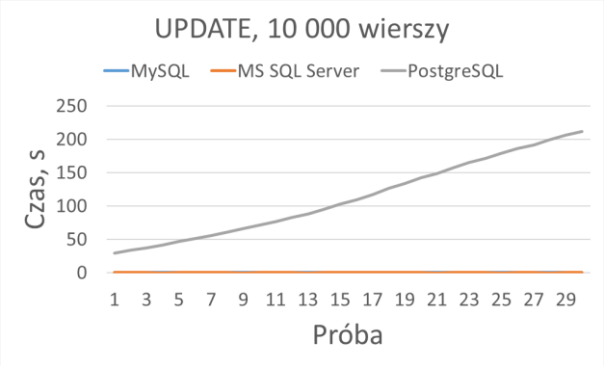


Rysunek 10: Wykres pudełkowy dla operacji aktualizacji 1 000 rekordów.

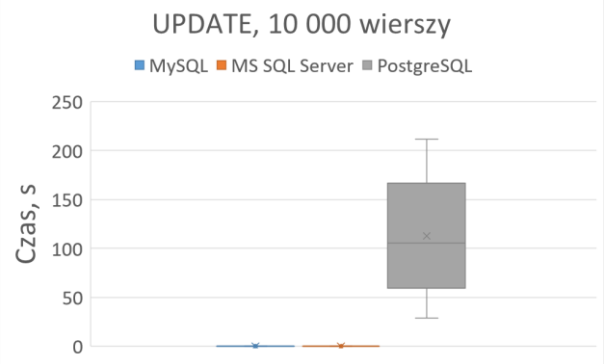
Kolejnym badanym przypadkiem była aktualizacja 10 000 wierszy. Tutaj wykres wyglądał podobnie do wykresów dla poprzednich przypadków (Rysunek 11). Powtórnie zwycięzcą okazał się MS SQL Server, MySQL zajął drugie miejsce, a PostgreSQL trzecie. Różnice w długościach czasów między systemem MS SQL Server a MySQL były niewielkie (około 0,2 s), a między MS SQL Server a PostgreSQL sięgały one kilkudziesięciu sekund. Wykres systemu PostgreSQL odznaczył się trendem wzrostowym, a w kwestii stabilności wyników najlepiej poradziły sobie MySQL i MS SQL Server (Rysunek 12).

Ranking wydajności dla aktualizacji rekordów był zbieżny dla trzech zestawów danych. MS SQL Server osiągał najlepsze wyniki, MySQL plasował się na drugiej pozycji, a PostgreSQL był na końcu. Ponadto, różnice w rezultatach były najmniejsze między MS SQL Server a MySQL, podczas gdy wyniki dla PostgreSQL

odstawały znacząco od konkurencji. Jeśli chodzi o stabilność uzyskanych wartości najlepsze okazały się systemy MS SQL Server i MySQL.



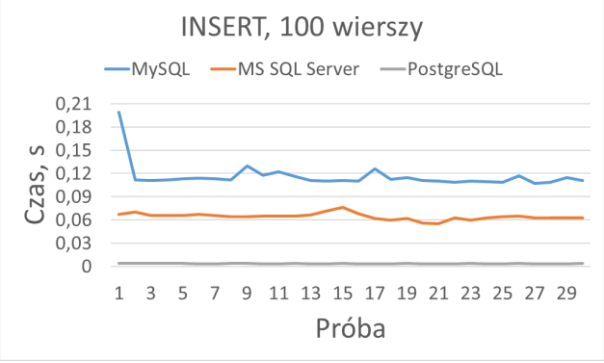
Rysunek 11: Wykres liniowy dla aktualizacji 10 000 rekordów.



Rysunek 12: Wykres pudełkowy dla aktualizacji 10 000 rekordów.

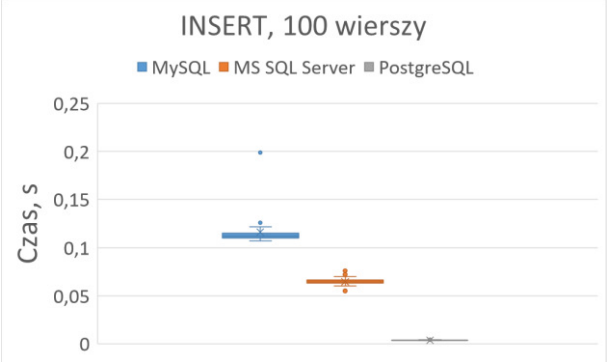
5.3. Wstawianie danych

Następnym badanym typem polecenia było wstawianie danych. Dla przypadku ze 100 wierszami (Rysunek 13) najlepsze rezultaty osiągnął PostgreSQL, a najgorsze MySQL. MS SQL Server zajął drugie miejsce. Różnice w wynikach były całkiem spore, ponieważ dla PostgreSQL i MS SQL Server oscylowały na poziomie 0,06 s, a dla PostgreSQL a MySQL wynosiły średnio 0,11 s. W kwestii stabilności rezultatów najlepszy okazał się PostgreSQL, podczas gdy MS SQL Server i MySQL cechowały większe dysproporcje w uzyskanych czasach (Rysunek 14).

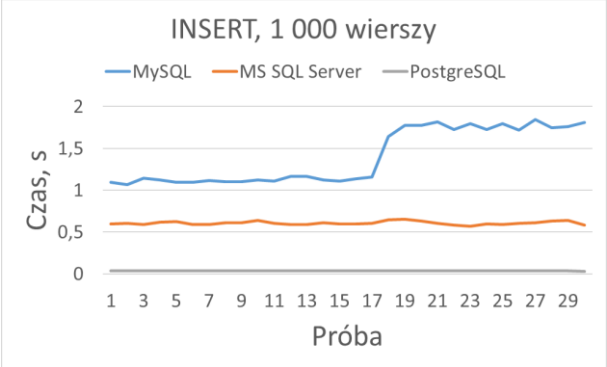


Rysunek 13: Wykres liniowy dla operacji wstawiania 100 rekordów.

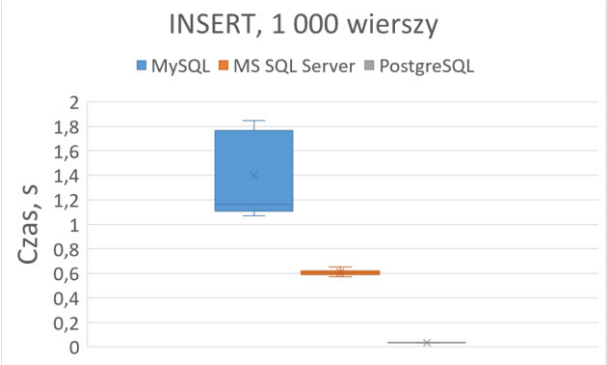
Po wstawieniu 1 000 rekordów miejsca na podium wyglądały identycznie jak poprzednio (Rysunek 15). PostgreSQL został zwycięzcą, MS SQL Server znalazł się zaraz po nim, a na końcu uplasował się MySQL. Różnice między wynikami ponownie były dość znaczące. Dla PostgreSQL i MS SQL Server były one równe około 0,57 s, a dla PostgreSQL i MySQL 1,36 s. Najlepszą stabilnością rezultatów mogły pochwalić się PostgreSQL i MS SQL Server (Rysunek 16). Natomiast MySQL po siedemnastej próbie zauważalnie wydłużył czasy wstawiania wierszy.



Rysunek 14: Wykres pudełkowy dla operacji wstawiania 100 rekordów.



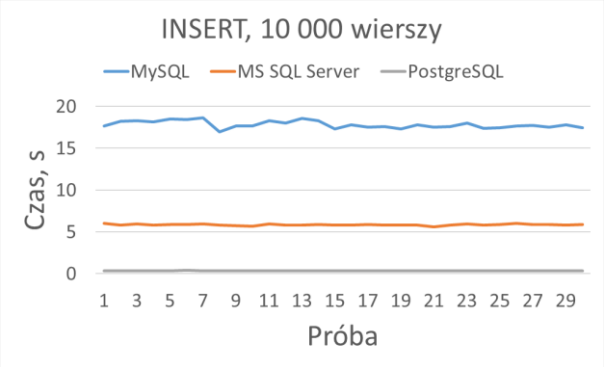
Rysunek 15: Wykres liniowy dla operacji wstawiania 1 000 rekordów.



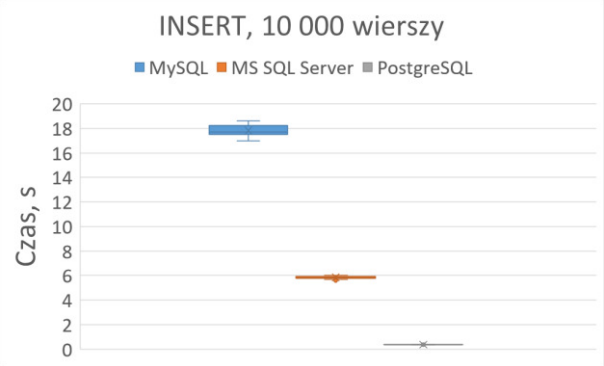
Rysunek 16: Wykres pudełkowy dla operacji wstawiania 1 000 rekordów.

Wstawienie 10 000 rekordów nie spowodowała zmian na podium (Rysunek 17). Najwydajniejszy znowu okazał się PostgreSQL, drugie miejsce zajął MS SQL Server, a trzecie MySQL. Różnice w wynikach między PostgreSQL a MS SQL Server były na poziomie około 5,48 s., a między PostgreSQL a MySQL wynosiły one około 17,83 s. W przypadku stabilności czasów

(Rysunek 18) powtórnie najlepsze były systemy PostgreSQL i MS SQL Server.



Rysunek 17: Wykres liniowy dla operacji wstawiania 10 000 rekordów



Rysunek 18: Wykres pudełkowy dla operacji wstawiania 10 000 rekordów.

Ranking systemów dla wstawiania wierszy wyglądał identycznie dla wszystkich trzech zestawów danych. PostgreSQL osiągał najkrótsze czasy, podczas gdy MS SQL Server i MySQL zajmowali odpowiednio drugie i trzecie miejsce. Różnice między wynikami były dość spore zarówno dla PostgreSQL i MS SQL Server jak i dla PostgreSQL i MySQL. Pod względem stabilności rezultatów najlepsze okazały się PostgreSQL i MS SQL Server.

6. Analiza i wyniki badania

W celu wyznaczenia najbardziej wydajnego systemu zarządzania bazami danych należało wykonać test statystyczny udowadniający istnienie statystycznie istotnych różnic między trzema zestawami danych. Zdecydowano się skorzystać z testu Kruskala-Wallisa [16] oraz internetowego narzędzia o nazwie „Online Web Statistical Calculators ..... for Categorical Data Analysis" [17], które pozwoliło to sprawdzić. Otrzymane rezultaty potwierdziły niezależność trzech prób od siebie dla wszystkich dziewięciu przypadków. Dla układu z zapytaniem SELECT i 100 wierszami (Tabela 3) wartość p-value wyniosła dużo mniej niż założony poziom istotności 0,05, dlatego odrzucono hipotezę zerową mówiącą o tym, że wszystkie próbki pochodzą z tego samego rozkładu na rzecz hipotezy alternatywnej, która zakładała, że jedna lub więcej próbek pochodzi z innego rozkładu. Dodatkowo, przeprowadzono testy post-hoc Dunna [18] w celu stwierdzenia pomiędzy którymi grupami pojawiły się istotne różnice (Tabela 4). Wszystkie wartości osiągnęły poziom niższy niż 0,05, zatem uznano, że wszystkie trzy zestawy danych są od siebie różne.

Tabela 3: Wynik testu Kruskala-Wallisa dla przypadku z zapytaniem SELECT i 100 wierszami

statystyka chi-kwadrat	stopnie swobody	wartość p-value
79,15	2	$6{,}5 \cdot 10^{-18}$

Tabela 4: Test post-hoc Dunna dla przypadku z zapytaniem SELECT i 100 wierszami

	MS SQL Server	MySQL
MySQL	0,000017	
PostgreSQL	0,000017	$1{,}73 \cdot 10^{-18}$

Wskazanie zwycięzcy dla każdego z przypadków opierało się na zliczeniu najkrótszych długości czasów dla wszystkich prób. System, który w większości prób osiągnął najlepsze wartości zostawał zwycięzcą dla danego przypadku. Uzyskane w ten sposób wyniki wykorzystano do utworzenia porównania wydajności systemów zarządzania bazami danych (Tabela 5). PostgreSQL okazał się być najszybszy aż w sześciu przypadkach, a MS SQL Server w trzech. MySQL nie zajął pierwszego miejsca ani razu. Tym samym system PostgreSQL został wskazany jako najwydajniejszy dla całego badania. Na tej podstawie należy odrzucić tezę pracy typującą MS SQL Server jako zwycięzcę. Rezultaty badania pozwoliły również odpowiedzieć na postawione pytania badawcze. Dla zapytania zwracającego i wstawiającego rekordy najszybszy był PostgreSQL, a dla zapytania aktualizującego wiersze MS SQL Server.

Tabela 5: Porównanie systemów bazodanowych

Przypadek	MySQL	MS SQL Server	PostgreSQL
SELECT, 100 wierszy	-	-	+
SELECT, 1 000 wierszy	-	-	+
SELECT, 10 000 wierszy	-	-	+
UPDATE, 100 wierszy	-	+	-
UPDATE, 1 000 wierszy	-	+	-
UPDATE, 10 000 wierszy	-	+	-
INSERT, 100 wierszy	-	-	+
INSERT, 1 000 wierszy	-	-	+
INSERT, 10 000 wierszy	-	-	+

7. Podsumowanie i wnioski

Celem pracy było przeprowadzenie analizy wydajnościowej trzech relacyjnych systemów zarządzania bazami danych: MySQL, PostgreSQL i Microsoft SQL Server. Badanie polegało na wykonaniu testów wykorzystujących wcześniej utworzone skrypty mierzące długości czasów operacji. Analizowane typy zapytań składały się z polecenia zwracającego dane (select), polecenia aktualizującego dane (update) i polecenia

wstawiającego dane (insert). Eksperyment przeprowadzono z użyciem trzech zestawów danych obejmujących kolejno 100, 1 000 i 10 000 rekordów. W testach uwzględniono dziewięć przypadków powstałych w oparciu o rodzaj polecenia i wariant zestawu danych. Schemat bazy danych zawierał trzynaście tabel i przedstawiał model dla sklepu internetowego zajmującego się technologiami konsumenckimi. W badaniu wykonano 35 prób dla każdego z przypadków, przy czym pierwsze pięć prób pominięto ze względu na kwestie optymalizacji bazy danych i kwestie przechowywania danych w pamięci podręcznej. Najwydajniejszym oprogramowaniem okazał się PostgreSQL. Osiągnął on najlepsze rezultaty dla sześciu przypadków. MS SQL Server wygrał tylko w trzech przypadkach, a MySQL nie zwyciężył ani razu. Odpowiedziano również na pytania badawcze postawione w pracy. PostgreSQL był najszybszy dla zapytania zwracającego i wstawiającego rekordy, a MS SQL Server dla zapytania aktualizującego wiersze.

Na podstawie przeprowadzonych badań sformułowano następujące wnioski:

1. PostgreSQL charakteryzował się bardzo dobrym poziomem wydajności i dobrą stabilnością dla operacji zwracania i wstawiania danych.
2. Microsoft SQL Server charakteryzował się bardzo dobrym poziomem wydajności dla operacji aktualizacji danych.
3. Microsoft SQL Server wyróżnił się dobrym poziomem wydajności i stabilności dla wszystkich testowanych rodzajów operacji.
4. MySQL cechowała zauważalna niestabilność w wydajności w przypadku zwracania większej ilości danych.
5. Nieprzerwane i wielokrotne aktualizowanie rekordów w systemie PostgreSQL spowodowało wydłużenie czasu wykonania tej operacji.

Badanie pozostawiło wiele obszarów, w których mogłoby znaleźć swój dalszy ciąg. Takie kwestie jak identyfikacja tzw. „wąskich gardeł" czy niewydajne indeksowanie to część niesprawdzonych aspektów wpływających na wydajność systemów zarządzania bazami danych. Ewentualna kontynuacja badania mogłaby pomóc wyjaśnić m.in. przyczyny niestabilności testowanych systemów oraz ich zachowania, a także pozwoliłaby znaleźć sposoby zwiększenia ich wydajności.

Literatura

[1] M. Grudzień, K. Korgol, D. Gutek, Porównanie możliwości wykorzystania oraz analiza wydajności baz danych na systemach mobilnych, praca magisterska, Politechnika Lubelska, Lublin, 2016.

[2] R. Kleweka, W. Truskowski, M. Skublewska-Paszkowska, Porównanie wydajności baz danych MySQL, MSSQL, PostgreSQL oraz Oracle z uwzględnieniem wirtualizacji, praca magisterska, Politechnika Lubelska, Lublin, 2020.

[3] K. Lachewicz, Analiza wydajności systemów bazodanowych: MySQL, MS SQL, PostgreSQL w kontekście aplikacji internetowych, praca magisterska, Politechnika Lubelska, Lublin, 2020.

[4] S. Stets, G. Kozieł, Analiza porównawcza wydajności baz danych pracujących pod kontrolą systemu Windows, praca magisterska, Politechnika Lubelska, Lublin, 2019.

[5] S. Kulshrestha, S. Sachdeva, Performance comparison for data storage - Db4o and MySQL databases, 2014 Seventh International Conference on Contemporary Computing (IC3) (2014) 166-170.

[6] R. Poljak, P. Pošcić, D. Jakšić, Comparative Analysis of the Selected Relational Database Management Systems, 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2017) 1496-1500.

[7] R. Kleweka, W. Truskowski, M. Skublewska-Paszkowska, Porównanie wydajności baz danych MySQL, MSSQL, PostgreSQL oraz Oracle z uwzględnieniem wirtualizacji, Journal of Computer Sciences Institute 16 (2020) 279-284.

[8] Y. Abubakar, Benchmarking popular open source RDBMS: a performance evaluation for IT professionals, International Journal of Advanced Computer Technology (IJACT) 3 (2014) 39-44.

[9] S. Tongkaw, A. Tongkaw, A comparison of database performance of MariaDB and MySQL with OLTP workload, 2016 IEEE Conference on Open Systems (ICOS) (2016) 117-119.

[10] M. -G. Jung, S. -A. Youn, J. Bae, Y. -L. Choi, A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment, 2015 8th International Conference on Database Theory and Application (DTA) (2015) 14-17.

[11] M. M. Eyada, W. Saber, M. M. El Genidy, F. Amer, Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments, IEEE Access 8 (2020) 110656-110668.

[12] H. Fatima, K. Wasnik, Comparison of SQL, NoSQL and NewSQL databases for internet of things, 2016 IEEE Bombay Section Symposium (IBSS) (2016) 1-6.

[13] M. Meekyung, Experiments of Search Query Performance for SQL-Based Open Source Databases, International Journal of Internet, Broadcasting and Communication 10 (2018) 31-38.

[14] R. Almeida, P. Furtado, J. Bernardino, Performance Evaluation MySQL InnoDB and Microsoft SQL Server 2012 for Decision Support Environments, Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering (2015) 56-62.

[15] Generator makiet danych „Mockaroo", https://www.mockaroo.com, [10.05.2023].

[16] W. H. Kruskal, W. A. Wallis, Use of Ranks in One-Criterion Variance Analysis, Journal of the American Statistical Association 47 (1952) 583-621.

[17] Internetowy kalkulator testów statystycznych „Online Web Statistical Calculators ..... for Categorical Data Analysis", https://astatsa.com/KruskalWallisTest/, [13.06.2023].

[18] O. J. Dunn, Multiple Comparisons Using Rank Sums, Technometrics 6 (1964) 241-252.