

Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework

Porównanie wydajności relacyjnych baz danych SQL Server, MySQL oraz PostgreSQL z zastosowaniem aplikacji webowej i frameworku Laravel

Rafał Wodyk*, Maria Skublewska-Paszowska

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

Abstract

Many database implementations are supported by application frameworks that can affect their performance. The paper presents a comparison of the performance of SQL Server, MySQL and PostgreSQL relational databases based on an application written in PHP using the Laravel framework. The time of performance for various types of queries, both simple and using column and table concatenation was evaluated. The obtained results for the same database structures differed depending on the operations performed on the databases. Looking at the entirety of the research conducted, it can be concluded that in the case of databases in which the number of records is not too large (up to 1000 records) and the technical parameters of the device on which the database is running are of low or medium class, MySQL performs very well.

Keywords: framework Laravel; SQL Server; MySQL; PostgreSQL

Streszczenie

Wiele implementacji baz danych jest wspieranych przez szkielety aplikacji, które mogą rzutować na ich wydajność. Artykuł przedstawia porównanie wydajności relacyjnych baz danych SQL Server, MySQL oraz PostgreSQL na podstawie aplikacji napisanej w języku PHP z wykorzystaniem frameworku Laravel. Pod uwagę wzięty został czas wykonania różnego typu zapytań, zarówno prostych, jak również z użyciem konkatenacji kolumn oraz tabel. Otrzymane wyniki zostały poddane wielowymiarowej analizie. Otrzymywane wyniki dla takich samych struktur baz danych różniły się w zależności od wykonywanych operacji na bazach. Patrząc na całokształt przeprowadzonych badań można dojść do wniosku, że w przypadku baz danych, w których liczba rekordów nie jest zbyt duża (do 1000 rekordów) oraz parametry techniczne urządzenia, na którym uruchomiona jest baza są klasy niskiej bądź średniej bardzo dobrze wypada MySQL.

Słowa kluczowe: framework Laravel; SQL Server; MySQL; PostgreSQL

*Corresponding author

Email address: rafal.wodyk@pollub.edu.pl (R. Wodyk), maria.paszowska@pollub.pl (M. Skublewska-Paszowska)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W obecnych czasach, gdzie z roku na rok coraz więcej spraw można załatwić bez wychodzenia z domu, kluczową rolę odgrywają rozwiązania bazodanowe, z których korzysta znaczna większość aplikacji zarówno internetowych, mobilnych, czy desktopowych. Wykorzystywane są one zarówno do przechowywania danych, które użytkownicy mogą zaobserwować na ekranach swoich urządzeń, jak i do rejestrowania zdarzeń, magazynowania wyników złożonych analiz oraz badań rynku.

Podczas tworzenia różnego typu projektów informatycznych najczęściej wybór pada na relacyjne mechanizmy bazodanowe [1]. Niejednokrotnie obecne systemy informatyczne przechowują bardzo duże ilości danych, co zarazem wymusza na administratorach jak najlepszą optymalizację szybkości ich działania m.in. przez optymalizację zapytań. Dlatego też bardzo ważna jest przemyślana struktura bazy, która w przypadku relacyjnych baz danych może mieć duży wpływ na jej wydajność i szybkość wykonywania zapytań, szczególnie tych rozbudowanych.

W przeszłości wiele osób skupiało się na przeprowadzaniu badań wydajności poszczególnych rozwiązań bazodanowych, zarówno SQL jak i NoSQL.

W artykule [2] autorzy porównują wydajności systemów RDMBS o licencji open source PostgreSQL oraz MySQL z komercyjnym rozwiązaniem Microsoft SQL Server. Badania polegały na porównaniu średnich czasów wykonania takich operacji jak SELECT, INSERT, UPDATE oraz DELETE dla każdego silnika bazodanowego. Każda operacja wykonywana była 30-40 razy zarówno synchronicznie jak i asynchronicznie. Do badań wykorzystano system operacyjny Ubuntu oraz SQL Server 2012, MySQL 5.5.31 i PostgreSQL 9.1.

W zapisie danych, zarówno dla małej liczby rekordów 1000 jak i dużej 100000, najlepiej radził sobie PostgreSQL, natomiast najwolniejszy okazał się MySQL. W przypadku aktualizacji danych najszybszym systemem bazodanowym okazał się SQL Server, a następnie po nim MySQL. Dla operacji kasowania rekordów, podobnie jak dla operacji aktualizacji najbardziej wydajnym rozwiązaniem okazał się SQL Server, lecz w tym przypadku najsłabszy był MySQL. Dla odczytu

danych najszybszą bazą okazał się PostgreSQL, a następnie po nim uplasował się MySQL. Autorzy tego artykułu na podstawie przeprowadzonych badań zaobserwowali, że najbardziej wydajnym systemem RDBMS spośród badanych jest PostgreSQL.

W tym artykule [3] autorzy analizują wydajności baz danych takich jak: Microsoft SQL Server, PostgreSQL, SQLite oraz MySQL na systemach mobilnych. Do badań wykorzystane zostały systemy mobilne Android w wersji 5.0.1 oraz Windows Mobile oraz serwer PHP wykorzystany do wystawiania odpowiedniego REST API. Badaniom poddane zostały operacje zapisu oraz odczytu danych. Badania zostały przeprowadzone z wykorzystaniem liczby rekordów od 10 do 1000000. W przypadku badań operacji zapisu rekordów o różnych typach najbardziej wydajnym systemem bazodanowym dla systemu mobilnego Windows Mobile okazał się PostgreSQL, a nieznacznie gorszy był SQL Server. Natomiast w przypadku systemu Android najwydajniejszym rozwiązaniem był SQL Server. Biorąc pod uwagę operację odczytu danych z użycie systemu Windows Mobile najlepszą wydajność zaprezentował SQL Server, lecz nieznacznie gorszy od niego był w tym badaniem przypadku MySQL. Na systemie Android wyjątkowo najszybszym rozwiązaniem okazał się MySQL. Autorzy analizowanego artykułu doszli do wniosku, że najszybszą formą wybierania oraz przechowywania danych jest SQL Server.

Analiza powyższych artykułów pokazała, że wyniki otrzymywane przez konkretne systemy bazodanowe są w dużej mierze zależne od technologii w jakiej powstała aplikacja oraz na jakim systemie operacyjnym działa. Niniejsza praca ma na celu przedstawienie wyników porównania wydajności najpopularniejszych relacyjnych baz danych takich jak MySQL, PostgreSQL oraz SQL Server. Są one wspierane przez obecnie najpopularniejszy szkielet aplikacji (ang. framework) dla języka PHP służący do tworzenia aplikacji internetowych, a mianowicie przez framework Laravel [4]. W dużej mierze dzięki swojej uniwersalnej składni interakcja z bazami danych jest niezwykle prosta, zarówno przy użyciu „surowego” języka SQL, konstruktora zapytań jak i Eloquent ORM (ang. object-relational mapping). Framework ten umożliwia generowanie takich samych struktur baz danych dla różnych wspieranych silników bazodanowych, co jest ułatwieniem zarówno w ewentualnej migracji z jednego rozwiązania na inne, jak i też do przeprowadzenia miarodajnych badań wydajnościowych. W niniejszym artykule otrzymane wyniki badań zostaną poddane wielowymiarowej analizie, dzięki czemu będzie można przekonać się, czy tak duża popularność silnika bazodanowego MySQL jest uzasadniona.

2. Relacyjne bazy danych

2.1. MS SQL

SQL Server jest to system do zarządzania relacyjnymi bazami danych, stworzony i licencjonowany przez firmę Microsoft [5]. Product SQL Server jest w większości przypadków rozwiązaniem płatnym wersje Standard oraz Enterprise. Istnieje również darmowa wersja

Express do zastosowania dla małych aplikacji. Głównym wykorzystywanym językiem zapytań w tej bazie danych jest Transact-SQL, który pozwala na używanie takich rozwiązań jak instrukcje warunkowe, czy pętle [5,6].

Pierwsza wersja produktu została wydana w 1989 roku jako SQL Server 1.0. Rozwiązanie to przez cały czas było rozwijane, a obecnie najnowsza wersja opatrzona numerem 15 pochodzi z 2019 roku. Obecnie SQL Server dostępny jest na systemy: Windows oraz Linux, a także na maszyny wirtualne jak np. Docker i Azure. Samo środowisko SQL Server jest skalowalne i może składać się z różnych elementów w zależności od potrzeb konkretnej aplikacji [7].

2.2. MySQL

MySQL jest darmowym - na licencji open source, systemem do zarządzania relacyjnymi bazami danych. Należy on do pakietu oprogramowania LAMP (Linux, Apache, MySQL, PHP) [8]. MySQL powstał w 1995 roku, a obecnie rozwijany jest przez firmę Oracle. System ten cechuje się elastycznością oraz dużą wydajnością (potrafi obsłużyć do miliona zapytań na sekundę). Nie bez przyczyny z tego rozwiązania korzystają takie serwisy jak YouTube, czy Facebook [9].

Jednym z mechanizmów poprawiający znacząco szybkość wykonywania zapytań jest zapisywanie do pamięci podręcznej ostatnich zapytań wraz z ich wynikami. W znaczącej mierze zmniejsza to czas zwrócenia żądanego wyniku.

Baza danych MySQL wspierana jest z wykorzystaniem API przez popularne języki programowania takie jak: PHP, Java, C#, C++, a bazę danych można uruchomić na wszystkich popularnych systemach operacyjnych jak: Linux, Windows, macOS [10].

2.3. PostgreSQL

PostgreSQL jest obiektowym systemem do zarządzania bazami danych (ORDBMS) na licencji open source. Podobnie jak MySQL, wspiera najpopularniejsze systemy operacyjne, m.in. Windows, Linux, macOS [11]. Początki rozwoju PostgreSQL sięgają 1973 roku, a od roku 1996 za sprawą społeczności open source był rozwijany i popularyzowany. Obecnie z omawianego systemu do zarządzania bazami danych korzysta m.in.: Apple, Cisco, Instagram, Spotify, czy Skype [9].

Różnego typu aplikacje wykorzystujące bazę danych PostgreSQL mogą być tworzone z wykorzystaniem popularnych języków programowania jak np. Java, Python, Ruby, .NET czy PHP i framework Laravel [12].

PostgreSQL jest zgodny z ACID (ang. atomicity, consistency, isolation, durability), wspiera transakcyjność oraz zawiera wbudowaną obsługę zwykłych indeksów B-drzewa i skrótów. Posiada aktualizowalne i zmateriaлизованe widoki, wyzwalacze i klucze obce. Obsługuje również funkcje (również niektóre NoSQL) oraz procedury składowane [13].

3. Metoda badań

W badanym przypadku pod uwagę brane były czasy wykonania zapytania dla trzech silników bazodanowych, tj. SQL Server, PostgreSQL oraz MySQL, w zależności od liczby rekordów w tabeli z ogłoszeniami. Liczba rekordów w tej tabeli mieściła się w przedziale od 1 do 20000, w kolejnych eksperymentach. Za każdym razem zwracane były wszystkie rekordy aktualnie znajdujące się w tabeli.

Do testów wykorzystany został komputer o następującej specyfikacji:

- procesor Intel i5-5200U 2,2GHz;
- 8 GB RAM DDR 3 1600MHz;
- dysk SSD Goodram IRDM 256GB SATA 3;
- system operacyjny Windows 10 64 bit.

Testom poddane zostały następujące bazy danych:

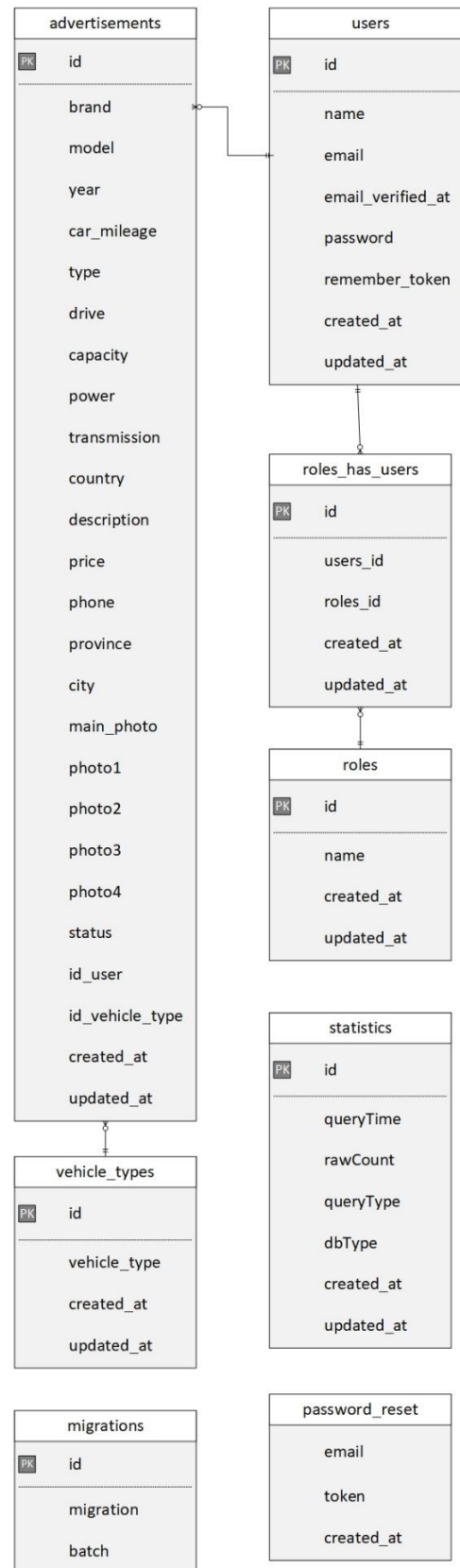
- SQL Server 2017;
- PostgreSQL 12;
- MySQL 8.0.

Bazy danych dla poszczególnych silników zostały wygenerowane z wykorzystaniem mechanizmu migracji wpieranego przez framework Laravel. Proces migracji polegał na stworzeniu struktury bazy danych na podstawie modelu bazy napisanego w języku PHP. Model każdej tabeli składa się ze scharakteryzowania każdej kolumny poprzez podanie jej nazwy, typu danych oraz nadania ewentualnych indeksów w postaci klucza podstawowego i obcego. Wykorzystując ten mechanizm utworzono takie same struktury baz dla badanych silników bazodanowych, jak przedstawiono na rysunku 1.

W badaniach skupiono się na najpopularniejszych operacjach bazodanowych, które można spotkać w większości aplikacji internetowych korzystających z baz danych czyli m.in. SELECT, INSERT, DELETE. Po każdym badanym zapytaniu następowało czyszczenie pamięci podręcznej cache w celu uzyskania jak najbardziej miarodajnych wyników. Czas wykonania instrukcji na bazie był mierzony z poziomu kodu PHP obliczając różnicę czasu po i przed wykonaniem instrukcji. Wyniki przeprowadzonych eksperymentów zostały zebrane i przedstawione w postaci wykresów, żeby w jak najlepszym stopniu zobrazować wydajność badanych systemów baz danych dla poszczególnych operacji. Całość badań została przeprowadzona z wykorzystaniem aplikacji internetowej w postaci portalu z ogłoszeniami motoryzacyjnymi, przechowującej dane dotyczące ogłoszeń i użytkowników w bazie danych.

Konfiguracja połączenia z bazą danych w projektach napisanych z wykorzystaniem frameworka Laravel znajduje się w pliku .env. Domyślną wykorzystywaną nazwą bazy danych współpracującej z aplikacją internetową wykorzystaną do badań było „web_db”. Jednym z najistotniejszych parametrów konfiguracyjnych baz danych wpływających na ich wydajność jest wielkość podstawowego bufora pamięciowego. Wykorzystywany jest on do buforowania danych podczas wykonywania

operacji odczytu i zapisu na bazie danych. W przypadku wszystkich trzech badanych silników bazodanowych jego wielkość była ustawiona na 128MB.



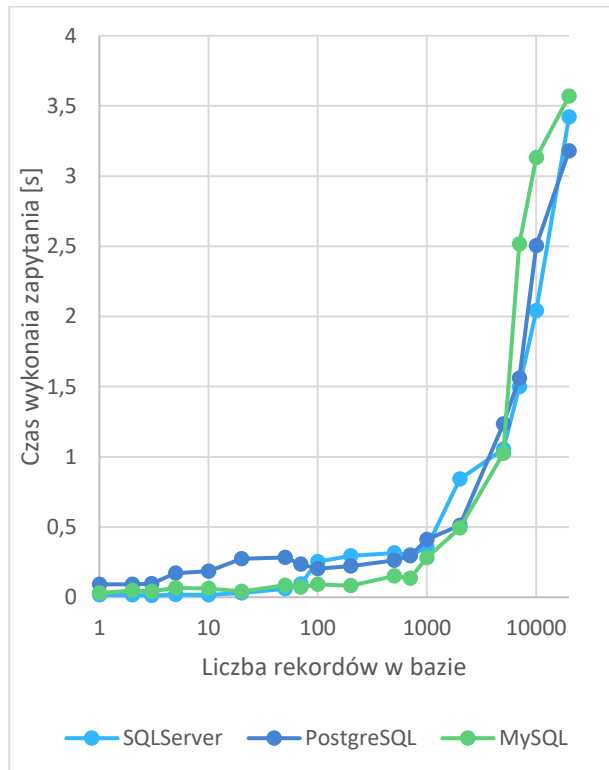
Rysunek 1: Schemat ERD bazy danych wykorzystywanej przez portal ogłoszeniowy

4. Wyniki badań

Najpopularniejszym typem zapytań bazodanowych jest instrukcja SELECT służąca do wybierania rekordów z tabel baz danych.

Na wykresie (Rys. 2) przedstawiono zebrane wyniki dla wszystkich badanych baz. W przypadku małej liczby rekordów (tj. do 1000 rekordów) w bazie danych, najwolniejszym rozwiązaniem okazywał się PostgreSQL, natomiast SQL Server i MySQL uzyskiwali czasy na zbliżonym poziomie. Większe dysproporcje można zauważyć w przypadku większej liczby rekordów w bazie (powyżej 5000), gdyż wtedy zdecydowanie najwolniejszym rozwiązaniem staje się popularny MySQL. Dla 20000 rekordów najszybsza okazała się baza danych PostgreSQL, będąc o prawie 0,2 s szybsza od SQL Server. W pewnym stopniu przewagę PostgreSQL można tłumaczyć tym, że SQL Server został stworzony z myślą o średnich i dużych aplikacjach, a co za tym idzie działania na sprzęcie bardziej wydajnym niż środowisko testowe użyte w tych badaniach.

W przypadku portali ogłoszeniowych bardzo często użytkownicy dostają możliwość skorzystania z wyszukiwarki ograniczającej wyniki do określonych fraz. W tym przypadku często wykorzystywana jest konkatenacja dwóch lub więcej kolumn, po których odbywa się wyszukiwanie. Dodatkowo niejednokrotnie wykorzystywane jest dodatkowe wyszukiwanie warunkowe z użyciem klauzuli WHERE, w celu dokładniejszego wyboru pasujących danych.

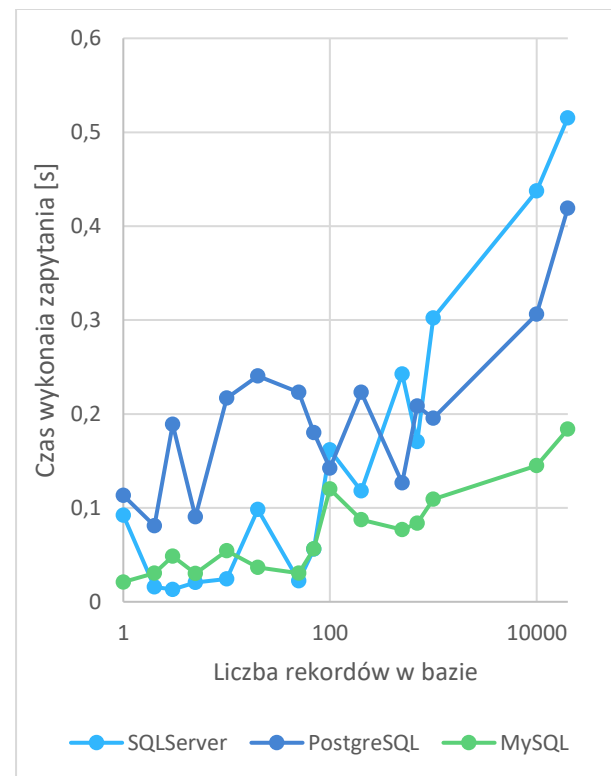


Rysunek 2: Czasy wykonania instrukcji Select w zależności od liczby zwróconych rekordów

W przypadku wykorzystywanej aplikacji również występuje wyszukiwarka, z wykorzystaniem której

przeprowadzono badania. Zapytania również jak poprzednio, wykonywane były na tabeli advertisements. Zwracane wyniki były zawężane do marki i modelu pojazdu oraz województwa, z którego pochodzi ogłoszenie. Brane były pod uwagę czasy wykonania instrukcji do wyżej opisanego wybierania danych. Otrzymane wyniki zostały przedstawione na rysunku 3.

Analizując otrzymane wyniki (Rys. 3) można zauważyć zależność podobną jak, w przypadku wybierania danych. Dla małej liczby rekordów (tj. do 500 rekordów) czasy wykonywania zapytań były zbliżone, choć na podstawie wykresu można zauważyć, że to jednak silnik bazodanowy MySQL był najszybszy. Jego przewaga w szybkości wykonywania zapytań jest tym bardziej zauważalna w przypadku większej liczby rekordów w bazie danych, gdzie wyniki, które osiąga są lepsze o ok. 0,2s w porównaniu z PostgreSQL oraz o ok. 0,3s w porównaniu z SQL Server.



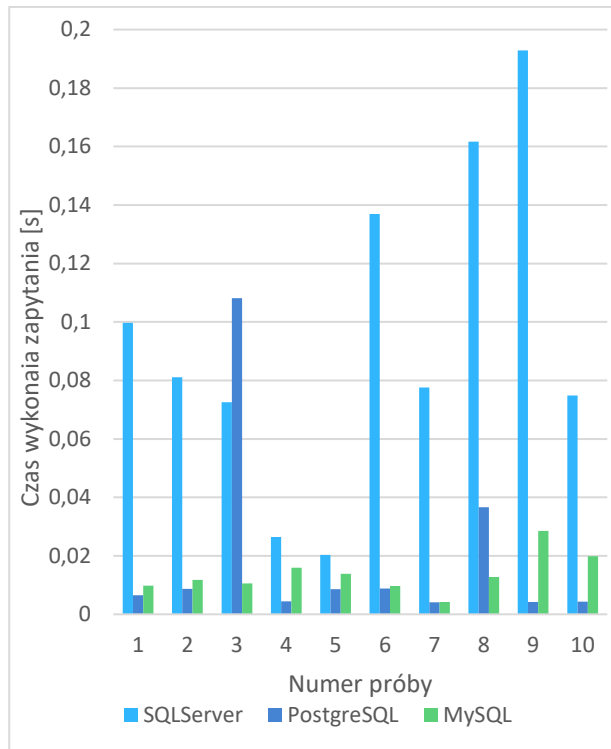
Rysunek 3: Czasy wykonania instrukcji Select w zależności od liczby rekordów w bazie

Kolejną z operacji poddanej badaniom było wstawianie nowych ogłoszeń motoryzacyjnych z poziomu interfejsu aplikacji. Badana operacja polegała na dodaniu jednego rekordu do tabeli advertisements oraz w przypadku braku danego typu pojazdu, jednego rekordu do tabeli vehicle_types. Dla każdego silnika bazodanowego przeprowadzono po 10 prób. Otrzymane wyniki zostały przedstawione na rysunku 4 i 5.

Na podstawie otrzymanych wyników można stwierdzić, że zdecydowanie najwolniejszym rozwiązaniem jest SQL Server, który jest średnio wolniejszy od swoich konkurentów o 0,08 s (Rys. 5). Jak duża jest to różnica świetnie prezentuje wykres (Rys. 4) na podstawie, którego od razu można zauważyć, że czasy SQL Server

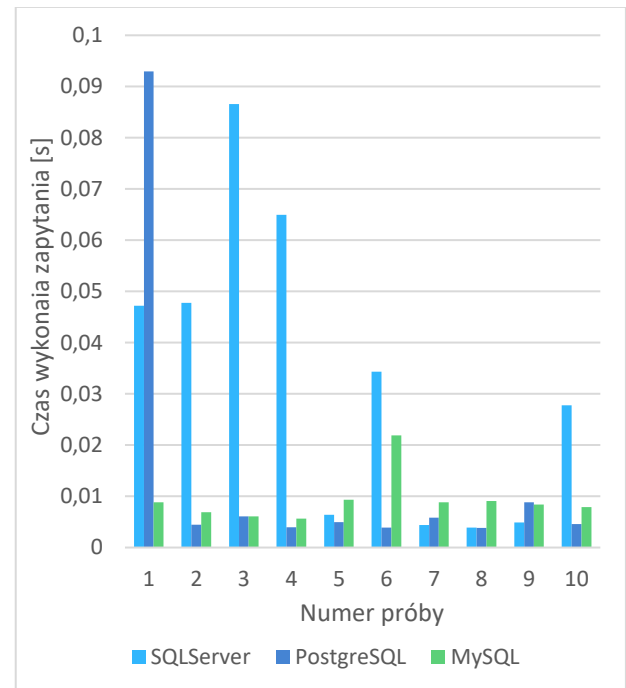
są kilkukrotnie gorsze, niż pozostałych baz danych. Najszybszym rozwiązaniem w tym badanym przypadku okazał się PostgreSQL, który okazywał się nieznacznie szybszy od MySQL.

Następną operacją, na której się skupiono podczas badań, była aktualizacja danych związanych z ogłoszeniami motoryzacyjnymi. Edycji zostało poddanych po 10 (dla każdej bazy danych) losowo wybranych ogłoszeń znajdujących się w tabeli advertisements.

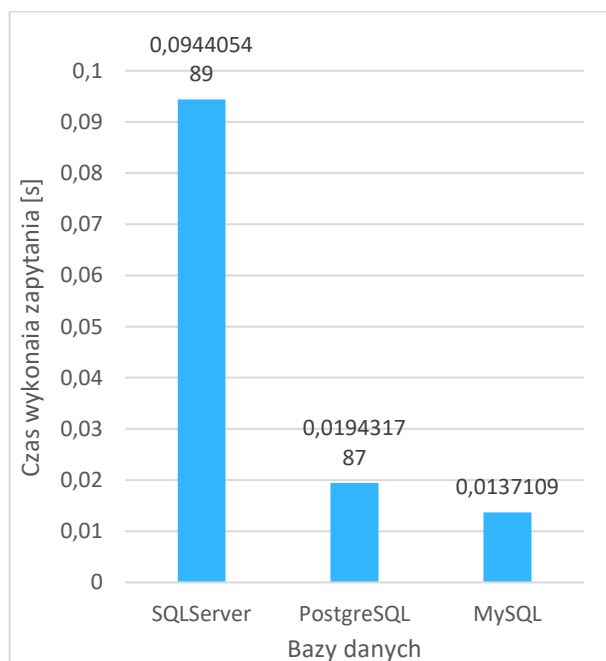


Rysunek 4: Czasy wykonania instrukcji Insert

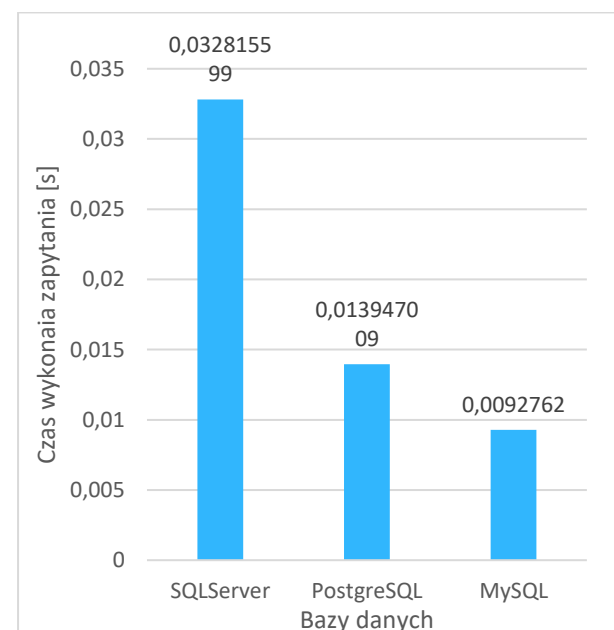
Porównując wyniki przedstawione na rysunku 6, z wynikami dotyczącymi operacji wstawiania danych, można zauważyć podobieństwo w kwestii różnicy czasów osiągniętych przez poszczególne systemy bazodanowe. Podobnie jak poprzednio, zdecydowanie najwolniejszym rozwiązaniem jest SQL Server, którego czasy zdecydowanie odstają od reszty co dobrze widać na rysunku 6. Również tak samo jak poprzednio, najszybszym rozwiązaniem okazał się PostgreSQL. Porównując średni czas na podstawie 10 prób można zauważyć, że różnica między SQL Server, a pozostałą dwójką jest mniejsza niż poprzednio i wynosi 0,02s (Rys. 7).



Rysunek 6: Czasy wykonania instrukcji Update



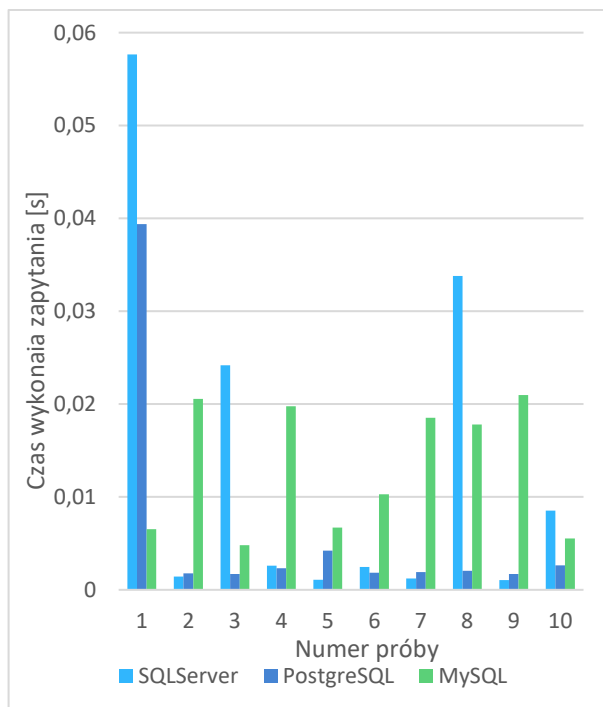
Rysunek 5: Średni czas wykonywania operacji dodawania danych



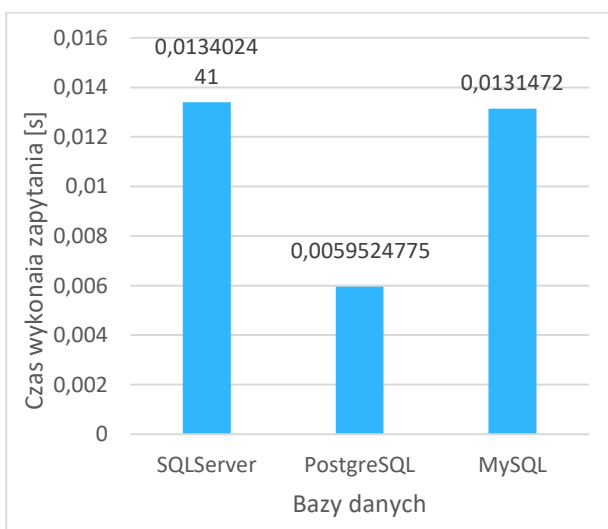
Rysunek 7: Średni czas wykonywania operacji aktualizacji danych

Ostatnią z operacji poddanych badaniom wydajności jest kasowanie danych. W tym przypadku pomiar czasu wykonania instrukcji DELETE był również oparty na danych znajdujących się w tabeli advertisements. Dokonano po 10 usunięć ogłoszeń motoryzacyjnych dla każdej z testowanych baz danych za pośrednictwem aplikacji internetowej.

Analizując otrzymane wyniki (Rys. 8) można zauważyć, że w tym przypadku najszybszym rozwiązaniem okazuje PostgreSQL. Jego średni czas wykonania operacji usunięcia jednego rekordu wynosi 0,00595s (Rys. 9) i jest lepszy od pozostałych dwóch rozwiązań o ok. 0,007s. SQL Server oraz MySQL w tym badaniu biorąc pod uwagę średnią arytmetyczną czasów na podstawie 10 prób osiągnęły bardzo zbliżone wyniki.



Rysunek 8: Czasy wykonania instrukcji Update



Rysunek 9: Średni czas wykonywania operacji kasowania danych

5. Wnioski

Otrzymywane wyniki dla takich samych struktur baz danych różniły się w zależności od wykonywanych operacji na bazach. Patrząc na całokształt przeprowadzonych badań można dojść do wniosku, że w przypadku baz danych, w których liczba rekordów nie jest zbyt duża (do 1000 rekordów) oraz parametry techniczne urządzenia, na którym uruchomiona jest baza są klasy niskiej bądź średniej bardzo dobrze wypada MySQL. Fakt ten potwierdzałby dużą popularność tego rozwiązania w przypadku małych konsumenckich aplikacji. Nieznacznie gorsze wyniki dla niezbyt dużej liczby danych (do 1000 rekordów) w bazie osiągał PostgreSQL, co daje mu drugie miejsce spośród trzech badanych rozwiązań.

Zaskoczeniem natomiast mogą być wyniki osiągnięte przez SQL Server, który w większości badanych operacji osiągał najgorsze czasy znacząco różniące się od konkurencji. Na plus natomiast można zapisać fakt, że w przypadku większej liczby rekordów w bazie danych (powyżej 1000 rekordów), jego czasy stawały się zbliżone do konkurencji. Może to potwierdzać założenie, że to rozwiązanie bazodanowe jest lepszym wyborem w przypadku średnich i dużych aplikacji, które operują na pokaźnych zasobach danych, a co za tym idzie wykorzystują urządzenia o wiele wydajniejsze niż testowe środowisko. Można się domyślać, że mając do dyspozycji wydajniejszy sprzęt SQL Server uzyskiwałby wyniki bardziej zbliżone do konkurencji.

Skupiając się natomiast na otrzymywanych wynikach w przypadku większej liczby rekordów (powyżej 1000) znajdujących się w bazie danych można zaobserwować, że MySQL traci pozycję lidera na rzecz PostgreSQL. Na podstawie tej obserwacji można stwierdzić, że najlepszym wyborem rozwiązania silnika bazy danych dla średniej bądź też dużej aplikacji jest właśnie PostgreSQL. Na jego korzyść z pewnością świadczy fakt, że jest oparty na licencji open source, a w dodatku jest cały czas wspierany i rozwijany, co z pewnością przemawia do sporej części twórców oprogramowania, żeby właśnie to rozwiązanie wybrać.

Literatura

- [1] Relacyjne bazy danych – dlaczego warto je znać?, https://teamquest.pl/blog/461_relacyjne-bazy-danych-dlaczego-warto-je-znac, [15.09.2020]
- [2] C. Truica, A. Boicea, Asynchronous Replication in Microsoft SQL Server, PostgreSQL and MySQL, International Conference on Cyber Science and Engineering, Guangzhou, 2013.
- [3] M. Grudzień, K. Korgol, D. Gutek, Porównanie możliwości wykorzystania oraz analiza wydajności baz danych na systemach mobilnych, Journal of Computer Sciences Institute 2 (2016) 133-139.
- [4] 8 Best PHP Frameworks in 2020, <https://athemes.com/collections/best-php-frameworks/>, [15.09.2020]

- [5] B. Masood-Al-Farooq, SQL Server 2014 Development Essentials, Packt Publishing, Birmingham, 2014.
- [6] SQL Server Tutorial, <https://www.sqlservertutorial.net>, [01.05.2020]
- [7] Środowisko MS SQL Server, <https://www.sqlpedia.pl/srodowisko-ms-sql-server/>, [01.05.2020]
- [8] MYSQL, <https://vavatech.pl/technologie/bazy-danych/mysql>, [26.04.2020]
- [9] PostgreSQL vs. MySQL: Which One Is Better for Your Use Case, <https://www.xplenty.com/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>, [26.04.2020]
- [10] Supported Platforms: MySQL Database, <https://www.mysql.com/support/supportedplatforms/data-base.html>, [16.10.2020]
- [11] S. Juba, A. Volkov, Learning PostgreSQL 11, Packt Publishing, Birmingham, 2019.
- [12] S. Riggs, G. Ciolli: PostgreSQL 9 Administration Cookbook, Second Edition, Packt Publishing, Birmingham, 2015.
- [13] PostgreSQL Vs. MySQL: Differences In Performance, Syntax, And Features, <https://blog.panoply.io/postgresql-vs.-mysql>, [26.04.2020]