

Analiza wydajności systemów bazodanowych: MySQL, MS SQL, PostgreSQL w kontekście aplikacji internetowych

Katarzyna Lachewicz*

Politechnika Lubelska, Katedra Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Głównym celem niniejszego artykułu jest weryfikacja, który z trzech systemów bazodanowych: MySQL, MS SQL, PostgreSQL, jest najbardziej wydajny w kontekście aplikacji internetowych. W artykule zawarto informacje dotyczące wykorzystanych systemów bazodanowych, jednakże kluczowym elementem artykułu są badania wydajności baz danych. Zostały one wykonane w oparciu o aplikację, której najważniejszym zadaniem jest wykonywanie zapytań do bazy danych. Program został zbudowany w oparciu o najnowsze technologie, takie jak framework Spring, biblioteka Hibernate oraz interfejs JDBC.

Słowa kluczowe: MySQL; MS SQL; PostgreSQL; wydajność baz danych

* Autor do korespondencji.

Adres e-mail: katarzyna.lachewicz@pollub.edu.pl

Performance analysis of selected database systems: MySQL, MS SQL, PostgreSQL in the context of web applications

Katarzyna Lachewicz*

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

Abstract. The main purpose of this article is to check which database: MySQL, MS SQL, PostgreSQL is the most efficient for Internet applications. This work contains information about the databases used, but the most important part of this article is database performance research. They are based on an application whose main task was database queries. The program was created based on new technologies, such as the Spring framework, the Hibernate library and JDBC Interface.

Keywords: MySQL; MS SQL; PostgreSQL; database performance

*Corresponding author.

E-mail address: katarzyna.lachewicz@pollub.edu.pl

1. Wprowadzenie

Rozpowszechnienie dostępu do Internetu umożliwiło korzystanie z ogromnych zasobów wiedzy i nowych funkcjonalności programów komputerowych. Powstała dzięki temu zupełnie nowa generacja programów, tzw. aplikacji webowych, które dzięki wykorzystaniu przeglądarki internetowej, oferowały analogiczne funkcjonalności jak aplikacje desktopowe. Zasadniczą zaletą aplikacji internetowych jest brak konieczności instalowania oprogramowania, a także dostęp do danych i funkcjonalności w dowolnym miejscu i czasie. Oprogramowanie, które wykorzystuje jako swoje środowisko Internet, stwarza możliwość przechowywania danych a także ich przetwarzania i modyfikowania [1]. Wybór takich rozwiązań wymaga gromadzenia znacznych ilości danych, co wiąże się z koniecznością stosowania różnego rodzaju bazy danych.

Wykorzystywanie baz danych w aplikacjach internetowych jest niezwykle istotne, gdyż dzięki nim, możliwa jest komunikacja serwera z użytkownikiem. Najczęściej odbywa się ona za pomocą formularzy, które pozwalają użytkownikowi dodawać oraz modyfikować dane. Ponadto możliwe jest również usuwanie i wyświetlanie odpowiednich zestawów danych. Warto zwrócić uwagę, iż na jakość aplikacji internetowych ogromny wpływ ma szybkość wykonywanych zapytań do bazy danych. Takie operacje decydują, czy użytkownik jest zadowolony z aplikacji, a co za

tym idzie czy będzie z niej korzystał. Opóźnione reakcje aplikacji, zbyt długie oczekiwanie na ładowanie się strony lub wykonywanie operacji w bazie danych może zniechęcić użytkownika, co grozi zastosowaniem innej, konkurencyjnej aplikacji. Wybór odpowiedniej bazy danych nie jest prosty, gdyż należy uwzględnić szereg czynników i parametrów takich jak szybkość wykonywania zapytań, metody implementacji bardziej skomplikowanych rozwiązań. O ostatecznym wyborze konkretnego systemu bazodanowego decyduje przeznaczenie aplikacji i przyjęte priorytety przy jej projektowaniu.

Obecnie, w dobie społeczeństwa informacyjnego oraz gospodarki opartej na wiedzy, ogromną rolę odgrywają metody gromadzenia i przetwarzania narastających każdego dnia coraz większych ilości danych. Analiza zgromadzonych danych jest źródłem wiedzy i rozwoju. Wydajność systemów bazodanowych odgrywa decydującą rolę w niemal wszystkich aspektach działania aplikacji internetowych. Zaniedbania i niedociągnięcia w tym zakresie skutkują chociażby tak niepożądanym wydłużeniem czasu ładowania strony czy też opóźnieniem potwierdzenia płatności. W komercyjnych zastosowaniach czas i wydajność posiadają decydujące znaczenie, dlatego też wybór optymalnej bazy danych decyduje o sukcesie danego serwisu.

2. Cel badań

Celem artykułu jest analiza wydajności trzech wybranych systemów bazodanowych w kontekście aplikacji internetowej. Porównanie wyników analizy pozwoli określić, który system bazodanowy działa najlepiej przy tego typu aplikacjach. Badania zostaną przeprowadzone według określonych scenariuszy, które prezentują sposób wykorzystania prezentowanych technologii.

Przeprowadzone badanie będzie wykonywane przy użyciu programu umożliwiającego symulację obciążenie serwera - JMeter. Dzięki powyższemu rozwiązaniu, możliwa będzie weryfikacja szybkości wykonywania zapytań dla poszczególnych baz danych. Przedmiotowe porównanie stanowić będzie istotną pomoc przy wyborze konkretnego systemu bazodanowego stosownie do przyjętych wcześniej priorytetów - szybkości bądź stabilności działania. Badania zostały oparte na aplikacji wykorzystującej bibliotekę Hibernate, interfejs JDBC oraz badane bazy danych.

Dla opisywanego artykułu, została sformułowana następująca hipoteza: „Systemy bazodanowe: MySQL, MS SQL i PostgreSQL, są jednakowo wydajne w kontekście aplikacji internetowych”. Poprawność zaprezentowanej tezy zostanie sprawdzona na podstawie wykonanych badań.

3. Przegląd literatury

Bazy danych pełnią coraz większą rolę w życiu każdego człowieka. Z uwagi na upowszechnienie się baz danych, istotne znaczenie nabiera ich wydajność. Dlatego też literatura naukowa dotycząca systemów bazodanowych, ich zastosowania, wdrożenia, jak również wydajności - jest liczna. Poniżej zaprezentowany został wybór prac naukowych o przedmiotowej tematyce.

W artykule „Comparative analysis of NoSQL (MongoDB) with MySQL Database” [2] autorstwa Lokesh Kumar, Shalini Rajawat oraz Krati Joshi, przedstawiono sposób zastąpienia relacyjnych baz danych przy wykorzystaniu nierelacyjnych systemów – NoSQL. Autorzy publikacji wykorzystali technologie takie jak MySQL oraz MongoDB, dokonując porównania ich wydajności a także łatwości zastosowania w aplikacjach z wykorzystaniem języka PHP. Podsumowując informacje zawarte w artykule należy wskazać, iż warto wykorzystywać nierelacyjne bazy danych w sytuacjach, gdy konieczne jest utworzenie aplikacji wykorzystującej bardzo duże ilości danych oraz wykonującej znaczną liczbę zapytań, takich jak wyszukiwanie czy operacje na danych.

Kolejną pracą naukową poświęconym opisywanej tematyce jest „Performance comparison for data storage - Db4o and MySQL databases” [3], której autorami są Sudhanshu Kulshrestha oraz Shelly Sachdeva. Autorzy przeprowadzili analizę porównawczą wydajności różnych systemów bazodanowych w aplikacjach dotyczących opieki zdrowotnej w Indiach. W pracy przeanalizowano trendy dotyczące systemów bazodanowych w opisywanym kraju, a następnie dokonano porównania bazy danych relacyjnej, takiej jak MySQL z obiektową – DB4o, pod względem wykorzystania przestrzeni dyskowej oraz trwałości danych. W rezultacie przeprowadzonych badań stwierdzono,

że krótszy czas zapisu danych uzyskała baza danych obiektowa – DB4o, jednocześnie wykorzystując więcej przestrzeni dyskowej od relacyjnej - MySQL.

Następną omawianą publikacją naukową jest „Performance Comparison of Relational Database with Object Database (DB4o)” [4], autorstwa K. E. Roopak, K. S. Swati Rao, S. Ritesh oraz Satyadhyan Chickerur. Artykuł opisuje badania z wykorzystaniem systemów bazodanowych takich jak Db4o i MySQL, na bardzo dużej bazie danych lotniska. Omówione zostały aspekty dotyczące pisania zapytań oraz wydajność prezentowanych baz danych. W wyniku przeprowadzonych badań uznano, iż w opisywanych warunkach – dla obszernej bazy danych lotniska - bardziej wydajnym systemem bazodanowym jest obiektowy system DB4o.

W artykule „A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment” [5], autorzy: Min-Gyue Jung, Seon-A Youn, Jayon Bae oraz Yong-Lak Choi, porównali wydajność baz danych relacyjnych i nierelacyjnych. Analizę porównawczą dokonano w oparciu o systemy PostgreSQL i MongoDB, przy wykorzystaniu bardzo dużych ilości danych. Ponadto w artykule opisano również projekt zwiększenia funkcjonalności dla nierelacyjnych baz danych. Podsumowując, przy nieustrukturyzowanym modelu danych, wykorzystanie MongoDB pozwala na odnotowanie większej wydajności. Natomiast przy ustrukturyzowanym modelu, PostgreSQL uzyskuje wyższą wydajność.

Artykuł „Analiza wydajności relacyjnych baz danych Oracle oraz MSSQL na podstawie aplikacji desktopowej” [6], której autorami są Grzegorz Dziewit, Jakub Korczyński oraz Maria Skublewska-Paszowska, zawiera porównanie wydajności baz danych Oracle i MS SQL w aplikacjach desktopowych. Co istotne, artykuł zawiera informacje dotyczące metodyki wykorzystywanej podczas porównywania relacyjnych systemów bazodanowych w zakresie średniego czasu wykonywania poszczególnych zapytań. Opracowany przez badaczy system badań wydajnościowych umożliwił określenie, który system jest lepszy w zależności od funkcjonalności aplikacji. W artykule udowodnione zostało, iż baza danych MS SQL lepiej wykonuje operacje INSERT oraz UPDATE, natomiast w przypadku zapytań wyszukiwania danych, krótszy czas wykonania otrzymał system Oracle.

Następny omawiany artykuł - „NoSQL real-time database performance comparison” [7], to praca autorstwa Diogo Augusto Pereira, Wagner Ourique de Moraes, Edison Pignaton de Freitas. W powyższym opracowaniu dokonano porównania cech, funkcjonalności oraz wydajności trzech nierelacyjnych systemów bazodanowych: Couchbase, MongoDB oraz RethinkDB. Testy obejmowały zarówno jedno jak i wielowątkowe zapytania. Zaprezentowane wyniki wskazują, że najlepszą wydajność uzyskał Couchbase, który zdobył najlepsze wyniki w większości przeprowadzonych testów.

Problematyki wydajności baz danych dotyczy również praca „A Comparison of Performance Between MSSQL Server and MongoDB for Telco Subscriber Data Management” [8], przygotowana przez Aaron Nichie,

Heung - Seo Koo. W publikacji ujęto porównanie wydajności bazy danych MS SQL i MongoDB, w kontekście wydajności w różnych fazach przepływu danych. W rezultacie prowadzonych badań stwierdzono, że baza MongoDB posiadała lepszą wydajność względem MS SQL.

Publikacja „Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage” [9], opracowana została przez Author Ken Ka-Yin Lee, Wai-Choi Tang, Kup-Sze Choi. W powyższej pracy ukazano badania naukowe bazujące na bazach danych NoSQL i XML, w celu zweryfikowania ich przydatności w przypadku danych klinicznych ustrukturyzowanych. Wykonane badania wskazują, iż baza nierelacyjna posiada lepsze wyniki pod względem szybkości wykonywania zapytań, natomiast baza XML wygrywa pod względem skalowalności, rozszerzalności i elastyczności.

Ostatnim omówionym artykułem jest „Performance Analysis for NoSQL and SQL” [10], którego autorami są Megha Katkar, Shah Kutchhi i Anchor Kutchhi. W powyższej pracy naukowej opisano badanie wydajności baz SQL i NoSQL. Wyniki przedstawione w artykule wskazują, iż spośród wszystkich systemów SQL i NoSQL najlepsze wyniki uzyskała baza danych nierelacyjna – FoundationDB. Dodatkowo autorzy zauważyli, że dla wszystkich systemów wyniki wydajności różnią się w zależności od typu operacji jaka miała być wykonana.

Wszystkie przedstawione powyżej artykuły naukowe odnoszą się do wydajności różnorodnych systemów bazodanowych, co wprost wskazuje na znaczne zainteresowania badaczy tą problematyką. Kwestie te są niezmiernie istotne, ponieważ obecnie niemal wszystkie aplikacje wykorzystują bazy danych. W konsekwencji ich wydajność jest kluczowym elementem do budowy aplikacji spełniającej wysokie wymagania użytkownika.

4. Opis obiektu badań

4.1. MySQL

MySQL to system bazodanowy rozwijany przez firmę Oracle, przeznaczony do zarządzania relacyjnymi bazami danych. Umożliwia przechowywanie, wyszukiwanie a także sortowanie i odczytywanie obszernych zbiorów danych. Co istotne, wskazane czynności może wykonywać nie jeden a wielu użytkowników jednocześnie - tzw. wielodostępność. Tą istotną cechą uzyskano dzięki zastosowaniu systemu zapytań, korzystającego z popularnego standardu języka SQL. Istnieje wiele rozszerzeń tego języka dodających nowe elementy i funkcjonalności. W konsekwencji wraz z rozwojem języka możliwe jest tworzenie coraz bardziej skomplikowanych zapytań.

Początki systemu MySQL datuje się na 1979 r. [11], jednakże publiczne udostępnienie nastąpiło w 1996 r. Oprogramowanie jest ciągle udoskonalane. Najnowsza wersja systemu została wydana 25 kwietnia 2019 roku.

4.2. MS SQL

MS SQL, czyli Microsoft SQL Server to system zarządzania bazami danych, rozpowszechniony przez

Microsoft. Jego zasadniczą zaletą jest rozszerzenie języka SQL – Transact-SQL. MS SQL jest systemem bazodanowym typu klient – serwer. W 1989 roku została wydana pierwsza wersja systemu - Ashton Tate/Microsoft SQL Server 1.0 [12]. Nad projektem pracowała firma Microsoft, Sybase oraz w późniejszym czasie Ashton Tate.

Najważniejszym elementem systemu jest silnik baz danych, którego zadaniem jest przechowywanie i przetwarzanie danych, a następnie zarządzanie nimi z wykorzystaniem modelu relacyjnego. W tym przypadku wykorzystywany jest system zarządzania danymi XML. Silnik bazodanowy umożliwia synchronizację i spójność danych [12].

Microsoft SQL Server umożliwia wykorzystanie operacji zgodnych z językiem zapytań SQL. Pozwala to na wykonywanie podstawowych operacji na bazach danych takich jak dodawanie nowych rekordów, wyświetlanie, edytowanie danych oraz ich usuwanie. Dodatkowo opisywany system umożliwia wykorzystanie bardziej skomplikowanych zapytań, dzięki wykorzystaniu Transact-SQL. Język ten rozszerza język SQL, dając programiście możliwość wykonywania bardziej skomplikowanych i złożonych instrukcji [12].

4.3. PostgreSQL

PostgreSQL to jeden z najpopularniejszych systemów bazodanowych umożliwiający zarządzanie relacyjnymi bazami danych. Prace rozpoczęto w 1977 roku w Berkeley na Uniwersytecie Kalifornijskim, a wstępna nazwa tego projektu to Ingres. W 1986 roku opublikowana została pierwsza wersja systemu. Po kilku latach zyskał on ogromną popularność, co umożliwiło dalsze rozwinięcie projektu.

Opisywana baza danych jest ciągle usprawniana, dzięki czemu stała się jednym z najbardziej funkcjonalnych, a zarazem niezawodnych systemów relacyjnych baz danych [13]. Opisywany system jest wciąż aktualizowany, co można zauważyć patrząc na najnowszą wersję PostgreSQL 11.5 wydaną w 8 sierpnia 2019 roku [14].

Jedną z najważniejszych cech PostgreSQL jest jego wielozadaniowość. Ponadto w razie konieczności wykonania niestandardowych procedur, możliwe jest programowe rozszerzanie za pomocą procedur przechowywanych [15].

5. Opis przebiegu badań

5.1. Opis aplikacji

Głównym celem artykułu jest sprawdzenie wydajności trzech systemów bazodanowych: MySQL, MS SQL oraz PostgreSQL. W konsekwencji kluczowe jest połączenie aplikacji z powyższymi bazami. Aplikacja składa się głównie z funkcji wykonywanych na bazach, dlatego też oprócz prostych instrukcji, dodane zostały również zapytania bardziej skomplikowane, dostosowane do celu badań.

Program został zbudowany w języku Java wykorzystując framework Spring, co umożliwia łatwiejsze tworzenie złożonych projektów aplikacji internetowych w języku Java. Dodatkowo wykorzystano bibliotekę Hibernate ułatwiającą dostęp do danych, poprzez zapewnienie odpowiedniej komunikacji między systemem bazodanowym a aplikacją

oraz interfejs JDBC pozwalający również na wykonywanie zapytań do bazy danych. Ponadto do badań wykorzystano narzędzia umożliwiające pracę z bazami danych jak phpMyAdmin, który umożliwia zarządzanie bazą danych MySQL. Dla MS SQL zastosowano Microsoft SQL Server Management Studio, a w przypadku PostgreSQL – pgAdmin.

W aplikacji zostały zrealizowane zapytania do bazy danych, takie jak wprowadzanie danych, ich wyświetlanie, usuwanie oraz edytowanie. Dodatkowo wykonane zostały bardziej skomplikowane zapytania, mające na celu sprawdzenie wydajności baz danych względem wykorzystanych komend SQL-owych. Ponadto zostały przetestowane różne rodzaje powiązania między tabelami a także wydajność baz danych z wykorzystaniem podzapytania.

5.2. Opis metody badawczej

W celu wykonania jak najdokładniejszych badań wydajnościowych baz danych, wybór scenariuszy testowych został wybrany w taki sposób, aby przetestowane zostały wszystkie podstawowe funkcjonalności bazodanowe wykorzystywane w aplikacjach internetowych. Wybrane operacje to między innymi wyświetlanie danych, dodawanie i edycja pojedynczych rekordów oraz usuwanie wybranych wiersz.

Tabela 1. Opis scenariuszy badania dla operacji na pojedynczych tabelach

Opis scenariusza	Tabela bazy danych	Liczba rekordów w tabeli	Powiązanie z innymi tabelami?
Wyszukiwanie pojedynczych rekordów	Wypożyczenia	6000	brak
Wyświetlanie zbioru danych	Książki	1000	brak
Wyświetlanie zbioru danych z komendą „is null”	Książki	1000	brak
Dodawanie pojedynczych rekordów	Wypożyczenia	6000	brak
Usuwanie pojedynczych rekordów	Wypożyczenia	6000	brak
Edycja danych wybranego rekordu	Książki	1000	brak

Wszystkie scenariusze zostały wykonane dla pojedynczych jak i dla powiązanych ze sobą tabel. Dlatego też scenariusze podzielono na dwie serie. Pierwsza dotyczy pojedynczej tabeli, natomiast druga powiązanych encji. Dodatkowo, warto zwrócić uwagę, iż wszystkie opisane poniżej scenariusze zostały wykorzystane przy badaniach w aplikacji internetowej z biblioteką Hibernate oraz interfejsem JDBS, a także w przypadku testów wykonanych bezpośrednio na bazie danych.

Weryfikacja wydajności bazy danych przy wykorzystaniu scenariuszy badań z obu serii pozwoli na określenie, które systemy są optymalne dla różnej złożoności zapytań. W tabelach 1 i 2 zawarto takie informacje jak: numer i opis scenariusza, tabele bazy danych jakie wykorzystywane zostały przy zapytaniu, liczba rekordów w każdej tabeli oraz

informacja czy wykorzystane zostało powiązanie między encjami.

Tabela 2. Opis scenariuszy badania dla operacji na powiązanych tabelach

Opis scenariusza	Tabela bazy danych	Liczba rekordów w tabeli	Powiązanie z innymi tabelami?
Wyszukiwanie z dwóch powiązanych tabel	Książki	1000	tak
	Kategorie	30	
Wyszukiwanie z czterech powiązanych tabel	Użytkownicy	500	tak
	Kategorie	30	
	Książki	1000	
	Wypożyczenia	6000	
Wyszukiwanie z podzapytaniem	Użytkownicy	500	tak
	Wydawnictwo	300	
	Książki	1000	
	Wypożyczenia	6000	
Wyszukiwanie z warunkiem i podzapytaniem zawierającym powiązane tabele	Użytkownicy	500	tak
	Kategorie	30	
	Książki	1000	
	Wypożyczenia	6000	
	Autorzy	300	
Wyszukiwanie z innym typem złączenia tabel	Wypożyczenia	6000	tak
	Książki	1000	
	Kategorie	30	
Wyszukiwanie z komendą „is null”	Książki	1000	tak
	Kategorie	30	

W przypadku badania wydajności aplikacji internetowej wykorzystano kryterium, według którego dla każdego scenariusza obliczony został średni czas wykonania danego zapytania oraz najkrótszy i najdłuższy czas odnotowany w badaniu. Ponadto dla każdego scenariusza wyliczono odchylenie standardowe oraz medianę.

5.3. Narzędzie wykorzystane do testowania

Do przeprowadzenia badań wydajnościowych wykorzystany został program Apache JMeter. Służy on do wykonywania testów funkcyjnych oraz obciążających, pozwalających określić maksymalną wielkość ruchu, który jest w stanie obsłużyć dany serwer. Program został zaprojektowany przez Stefano Mazzocchi z Apache Software Foundation w celu testowania wydajności Apache Jserv [16]. Następnie narzędzie zostało przeprojektowane oraz wyposażone w interfejs graficzny, co umożliwiło stosowanie go do testów funkcyjnych.

JMeter pozwala na przetestowanie zasobów takich jak pliki, bazy danych, serwery FTP oraz języki umożliwiające programowania aplikacji internetowych: Java, PHP, ASP.NET [16]. Umożliwia również symulowanie ruchu na serwerze oraz analizowanie wydajności pod wpływem obciążenia.

6. Prezentacja rezultatów badań

W celu wykonania jak najdokładniejszych badań, każdy scenariusz przeprowadzony z wykorzystaniem aplikacji internetowej, został powtórzony 1000 razy. Dzięki temu możliwe było uzyskanie bardziej precyzyjnych średnich czasów wykonania zapytania.

6.1. Badania na pojedynczych tabelach z wykorzystaniem aplikacji z biblioteką Hibernate

W tabeli 3 zestawiono wyniki średnich czasów dla poszczególnych scenariuszy. Każde badanie zostało wykonane dla trzech systemów bazodanowych: MySQL, MS SQL i PostgreSQL.

Tabela 3. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na pojedynczej tabeli

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie pojedynczych rekordów	5,2 ms	5,2 ms	4,7 ms
Wyszukiwania zbioru danych	5,3 ms	5,7 ms	6,5 ms
Wyświetlanie zbioru danych z komendą „is null”	6,2 ms	6,0 ms	5,8 ms
Dodawanie pojedynczych rekordów	8,3 ms	8,4 ms	7,0 ms
Usuwanie pojedynczych rekordów	6,3 ms	6,9 ms	6,8 ms
Edycja danych wybranego rekordu	7,3 ms	7,4 ms	7,7 ms

Dane z tabeli 3 wskazują, iż w przypadku prostych zapytań na bazie danych, czyli takich, które wykonywane są na pojedynczej tabeli, wyniki są bardzo zbliżone dla zaprezentowanych baz danych. MySQL uzyskał najniższe wyniki w przypadku wyszukiwania pojedynczych rekordów, zbiorów danych oraz usuwania i edytowania pojedynczych danych. Natomiast system bazodanowy PostgreSQL zdobył najlepszy wynik dla scenariusza wyświetlania zbioru danych z wykorzystaniem komendy SQL „is null” oraz dla dodawania pojedynczych rekordów

Warto nadmienić, iż dane przedstawione dla pojedynczej tabeli są bardzo zbliżone dla wszystkich systemów bazodanowych. Dodatkowo, średnie czasy uzyskane dla przedstawionych scenariuszy nie przekraczają 8,5 ms, co oznacza optymalną odpowiedź serwisu bazodanowego.

6.2. Badania na tabelach powiązanych z wykorzystaniem aplikacji z biblioteką Hibernate

Kolejny zestaw badań polega na wykonywaniu zapytań wyszukiwania danych z wykorzystaniem powiązania tabel w aplikacji z wykorzystaniem biblioteki Hibernate. W tym przypadku sprawdzona została wydajność baz danych przy wykorzystaniu zapytań zawierających instrukcje warunkowe, podzapytania oraz powiązanie z większą ilością tabel. Wyniki dla wszystkich scenariuszy zostały zaprezentowane w tabeli 4.

Najdłuższe średnie czasy wykonywania zapytań zostały odnotowane w przypadku scenariusza wyszukiwania danych z czterech powiązanych tabel. W tym przypadku baza danych MySQL uzyskała wynik 114,3 ms, MS SQL – 21,0 ms a PostgreSQL 16,5 ms. Z powyższego zestawienia wynika, iż najlepiej poradziła sobie baza danych PostgreSQL.

Natomiast najkrótsze czasy odnotowane zostały dla wyszukiwania danych z instrukcją warunkową „is null”. W przypadku bazy MySQL wyliczony został średni czas 7,4 ms, dla MS SQL – 5,5 ms, a PostgreSQL – 4,2 ms. W tym przypadku również system bazodanowy PostgreSQL uzyskał

najlepsze wyniki, jednakże różnice względem pozostałych baz danych są niewielkie.

Tabela 4. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na tabelach powiązanych

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie z dwóch powiązanych tabel	13,3 ms	6,4 ms	6,1 ms
Wyszukiwanie z czterech powiązanych tabel	114,3 ms	21,0 ms	16,5 ms
Wyszukiwanie z podzapytaniem	51,5 ms	19,8 ms	15,3 ms
Wyszukiwanie z warunkiem i podzapytaniem zawierającym powiązane tabele	86,1 ms	12,6 ms	7,4 ms
Wyszukiwanie z innym typem złączenia tabel	45,8 ms	6,2 ms	4,7 ms
Wyszukiwanie w komendą „is null”	7,4 ms	5,5 ms	4,2 ms

Warto zwrócić uwagę, iż w przypadku wszystkich scenariuszy z serii badań wykonanych na powiązanych tabelach, najkrótsze średnie czasy uzyskała baza danych PostgreSQL. Dodatkowo warto nadmienić, iż w tym przypadku odnotowano wszystkie średnie czasy dla tego systemu bazodanowego poniżej 17 ms, co jest optymalnym wynikiem.

W zaprezentowanych badaniach w tabeli 4 najgorzej poradziła sobie baza danych MySQL, której średnie czasy dochodzą do wartości 114,3 ms. Wyniki te są dużo wyższe w porównaniu do pozostałych baz danych. W przypadku systemu MS SQL, najdłuższy średni czas to 21,0 ms, co oznacza, iż opisywana baza również uzyskuje lepsze wyniki niż MySQL.

Podsumowując, z badaniami opartymi na powiązanych tabelach najlepiej poradził sobie system bazodanowy PostgreSQL, który uzyskał najniższe wyniki we wszystkich testach opisanych w tabeli 4.

6.3. Badania na pojedynczych tabelach z wykorzystaniem aplikacji z interfejsem JDBC

Badania zaprezentowane w tabeli 5 przedstawiają wyniki zapytań wykonanych na pojedynczej tabeli z wykorzystaniem aplikacji internetowej z interfejsem JDBC.

Tabela 5. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na pojedynczej tabeli

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie pojedynczych rekordów	4,3 ms	4,7 ms	3,8 ms
Wyszukiwania zbioru danych	3,8 ms	4,1 ms	3,3 ms
Wyświetlanie zbioru danych z komendą „is null”	3,6 ms	3,8 ms	3,0 ms
Dodawanie pojedynczych rekordów	3,8 ms	5,2 ms	4,4 ms
Usuwanie pojedynczych rekordów	3,2 ms	5,4 ms	2,9 ms
Edycja danych wybranego rekordu	3,3 ms	4,8 ms	3,9 ms

Analizując zaprezentowane dane można zauważyć, iż baza PostgreSQL najlepiej poradziła sobie w przypadku wyszukiwania pojedynczych rekordów, zbiorów danych a także wyszukiwania z wykorzystaniem komendy „is null” oraz usuwaniem pojedynczych rekordów. Natomiast w przypadku dodawania i edycji pojedynczych danych, najkrótsze średnie czasy odnotowano dla systemu MySQL.

Wyniki zaprezentowane w tabeli 5 dla wszystkich systemów są bardzo zbliżone i nie przekraczają 5,2 ms. Jednak warto zwrócić uwagę, iż najdłuższe średnie czasy we wszystkich scenariuszach uzyskała baza MS SQL.

6.4. Badania na tabelach powiązanych z wykorzystaniem aplikacji z interfejsem JDBC

W tabeli 6 zaprezentowane zostały wyniki badania wykonane na aplikacji z interfejsem JDBC dla tabel powiązanych.

Baza danych, która uzyskała najkrótsze średnie czasy dla wszystkich scenariuszy to PostgreSQL. Czasy wykonania zapytań mieszczą się w zakresie od 2,9 ms do 7,2 ms. Kolejną bazą, która uzyskała wyniki od 4,1 ms do 12,8 ms to MS SQL. Natomiast najgorzej poradziła sobie baza MySQL z wynikami od 4,3 ms do 81,1 ms.

Tabela 6. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na tabelach powiązanych

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie z dwóch powiązanych tabel	8,4 ms	4,8 ms	3,3 ms
Wyszukiwanie z czterech powiązanych tabel	81,1 ms	12,8 ms	6,6 ms
Wyszukiwanie z podzapytaniem	38,3 ms	12,4 ms	7,2 ms
Wyszukiwanie z warunkiem i podzapytaniem zawierającym powiązane tabele	68,4 ms	11,2 ms	4,9 ms
Wyszukiwanie z innym typem złączenia tabel	47,2 ms	6,0 ms	4,3 ms
Wyszukiwanie w komendą „is null”	4,4 ms	4,1 ms	2,9 ms

Zapytania, które uzyskały najdłuższe średnie czasy realizacji to wyszukiwanie z czterech powiązanych tabel oraz wyszukiwanie z podzapytaniem. Warto również zauważyć, iż wyniki w opisanym badaniu są gorsze niż w przypadku wykorzystania Hibernate. Oznacza to, iż wykorzystanie tej biblioteki powoduje wykonywanie dodatkowych operacji w tle, a tym samym zmniejszana jest wydajność systemu.

6.5. Badania na tabelach powiązanych wykonane bezpośrednio na bazie danych

Tabela 7 zawiera wyniki badań wykonanych bezpośrednio na bazach danych, na których dodatkowo zastosowano czyszczenie pamięci cache. Testy wykonane zostały z wykorzystaniem scenariuszy jak w poprzednich punktach. Każde badanie zostało powtórzone 10 razy.

Po analizie uzyskanych danych można stwierdzić, iż po wyczyszczeniu pamięci cache najdłuższe średnie czasy uzyskała baza PostgreSQL. Wyniki wahają się od 65,9 ms do

283,5 ms. Natomiast najkrótsze czasy w większości zapytań uzyskała baza MS SQL z wynikami od 10,6 ms do 41,7 ms.

Tabela 7. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na pojedynczej tabeli

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie pojedynczych rekordów	18 ms	15,3 ms	261,5 ms
Wyszukiwania zbioru danych	12,1 ms	41,7 ms	283,5 ms
Wyświetlanie zbioru danych z komendą „is null”	18,2 ms	18,1 ms	258,2 ms
Dodawanie pojedynczych rekordów	56,5 ms	10,6 ms	74,3 ms
Usuwanie pojedynczych rekordów	34,1 ms	15,3 ms	80,3 ms
Edycja danych wybranego rekordu	40,5 ms	15,6 ms	65,9 ms

6.6. Badania dla tabel powiązanych wykonane bezpośrednio na bazie danych

Wyniki badań jakie uzyskano dla bazy danych w przypadku powiązanych encji zawarte zostały w tabeli 8. Wskazują one, iż najkrótsze średnie czasy dla wszystkich zapytań uzyskała baza MS SQL z wynikami od 15 ms do 76,1 ms. System bazodanowy MySQL uzyskał lepsze wyniki od PostgreSQL w przypadku zapytań wyszukiwania z dwóch powiązanych tabel, z wykorzystaniem podzapytania oraz z zastosowaniem komendy „is null”. Natomiast dla PostgreSQL odnotowano niższe wyniki dla wyszukiwania z wykorzystaniem czterech powiązanych tabel, z zastosowaniem innych typów złączeń oraz wyszukiwanie z podzapytaniem i powiązanymi encjami.

Tabela 8. Wyniki średnich czasów dla poszczególnych scenariuszy w przypadku testów na tabelach powiązanych

Scenariusz	MySQL	MS SQL	PostgreSQL
Wyszukiwanie z dwóch powiązanych tabel	114,6 ms	47,7 ms	262,7 ms
Wyszukiwanie z czterech powiązanych tabel	602,8 ms	39,9 ms	264,7 ms
Wyszukiwanie z podzapytaniem	285,6 ms	34,3 ms	302 ms
Wyszukiwanie z warunkiem i podzapytaniem zawierającym powiązane tabele	480,8 ms	76,1 ms	306,3 ms
Wyszukiwanie z innym typem złączenia tabel	443,7 ms	27,5 ms	305,4 ms
Wyszukiwanie w komendą „is null”	35,1 ms	19,2 ms	304,9 ms

Na podstawie wyników przeprowadzonych badań należy stwierdzić, że po oczyszczeniu pamięci cache, najbardziej wydajnym systemem bazodanowym jest MS SQL. Warto zwrócić również uwagę, iż MySQL radzi sobie lepiej w porównaniu z PostgreSQL w przypadku prostszych zapytań, natomiast baza danych PostgreSQL w bardziej złożonych.

7. Wnioski

Głównym celem niniejszego artykułu było przeprowadzenie badań wydajności trzech systemów

bazodanowych a także określenie, który z nich najlepiej wykorzystać podczas budowy aplikacji internetowych.

Analizując dane zaprezentowane w rozdziale „Badania”, należy wskazać, iż w przypadku aplikacji internetowych, najbardziej wydajnym systemem bazodanowym jest PostgreSQL. Przedmiotowa baza danych zdecydowanie lepiej radzi sobie z bardziej skomplikowanymi zapytaniami, które wykorzystują powiązanie tabel, zarówno dla aplikacji wykorzystującej bibliotekę Hibernate jak i interfejs JDBC. Warto również nadmienić, iż najdłuższe średnie czasy realizacji zapytań jakie uzyskano dla tej bazy danych to 16,5 ms, przy czym większość zapytań była realizowana w czasie poniżej 8 ms. Wyniki te są dużo korzystniejsze niż w przypadku pozostałych systemów bazodanowych. Natomiast najgorzej z przedstawionych baz danych poradził sobie MySQL, którego średnie czasy wykonywania zapytań dochodziły do 114,3 ms. W przypadku zapytań wykorzystujących pojedynczą tabelę, średnie czasy opisywanych systemów są bardzo do siebie zbliżone. Warto jednak zwrócić uwagę, iż w przypadku aplikacji internetowych często wykorzystywane są bardziej skomplikowane zapytania. Dodatkowo w artykule udowodniono, iż lepszą wydajność systemów bazodanowych uzyskała aplikacja, która wykorzystywała interfejs JDBC. Średnie czasy dla wszystkich zapytań były niższe, od tych z zastosowaniem biblioteki Hibernate.

Ponadto, wykonane zostały badania na bazach danych z wykorzystaniem czyszczenia pamięci cache, w celu zweryfikowania, który z systemów bazodanowych jest najbardziej wydajny. W wyniku badań wykonanych bezpośrednio na bazach danych, optymalnym systemem okazał się MS SQL. Otrzymany rezultat wskazuje na to, iż opisana baza danych w mniejszym stopniu bazuje na danych przechowywanych w pamięci podręcznej, w porównaniu do pozostałych. Takie rozwiązanie wynika ze sposobu optymalizacji zapytań jaki został wykorzystany przy budowie wybranego systemu bazodanowego.

Wydajność baz danych jest niezwykle istotnym aspektem podczas tworzenia programu, ponieważ to ona decyduje o szybkości działania aplikacji, co w konsekwencji przekłada się na zadowolenie klientów. Badania przeprowadzone w ramach artykułu, mają na celu weryfikację wydajności kilku najbardziej popularnych systemów bazodanowych pod kątem zastosowań w aplikacji internetowych.

Teza, która została postawiona w artykule: „Systemy bazodanowe: MySQL, MS SQL i PostgreSQL, są jednakowo wydajne w kontekście aplikacji internetowych” została obalona. W wyniku analizy badań przeprowadzonych na aplikacji z wykorzystaniem biblioteki Hibernate oraz interfejsu JDBC, najbardziej wydajnym systemem okazał się PostgreSQL.

Odpowiedni dobór technologii znacznie usprawnia działanie programu, co z kolei może być kluczowe dla klientów i tym samym przesądzić o powodzeniu biznesowym aplikacji. W przypadku aplikacji internetowych, najbardziej wydajnym systemem bazodanowym okazał się PostgreSQL. System ten sprawnie i efektywnie radzi sobie z zarówno prostszymi jak i bardziej rozbudowanymi zapytaniami do bazy danych, co wskazuje na jego szczególną przydatność w zastosowaniach związanych z tworzeniem aplikacji internetowych.

Literatura

- [1] Marek Miłoś (red.): Aplikacje internetowe – od teorii do praktyki, 2018.
- [2] Lokesh Kumar, dr. Shalini Rajawat, Krati Joshi: Comparative analysis of NoSQL (MongoDB) with MySQL Database, 2015.
- [3] Sudhanshu Kulshrestha, Shelly Sachdeva, Performance comparison for data storage - Db4o and MySQL databases, 2014.
- [4] Roopak K.E., Swati Rao K.S., Ritesh S., Satyadhyam: Performance Comparison of Relational Database with Object Database (DB4o), 2013
- [5] Min-Gyue Jung, Seon-A Youn, Jayon Bae, Yong-Lak Choi: A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment, 2015.
- [6] Grzegorz Dziewit, Jakub Korczyński, Maria Skublewska-Paszowska: Analiza wydajności relacyjnych baz danych Oracle oraz MSSQL na podstawie aplikacji desktopowe, 2018.
- [7] Diogo Augusto Pereira, Wagner Ourique de Moraes, Edison Pignaton de Freitas: NoSQL real-time database performance comparison, 2017.
- [8] Aaron Nichie, Heung-Seo Koo: A Comparison of Performance Between MSSQL Server and MongoDB for Telco Subscriber Data Management, 2016.
- [9] Ken Ka-Yin Lee, Wai-Choi Tang, Kup-Sze Choi: Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage, 2013.
- [10] Megha Katkar, Shah and Anchor Kutchhi: Performance Analysis for NoSQL and SQL, 2015.
- [11] MySQL, <http://vavatech.pl/technologie/bazy-danych/mysql> [30.08.2019].
- [12] Adam Pelikant, MS SQL Server. Zaawansowane metody programowania, 2014.
- [13] PostgreSQL, <http://vavatech.pl/technologie/bazy-danych/postgresql> [30.08.2019].
- [14] PostgreSQL 12 Released!, <https://www.postgresql.org/docs/11/release-11-5.html> [30.09.2019].
- [15] Porównanie relacyjnych SZBD: SQLite, MySQL, PostgreSQL, <https://hostovita.pl/blog/porownanie-relacyjnych-systemow-zarządzania-bazami-danych-sqlite-mysql-postgresql/> [30.08.2019].
- [16] JMeter – narzędzie testera, <http://2016.testwarez.pl/jmeter-narzedzie-testera/> [10.09.2019].