

# A comparative analysis of non-relational databases in e-commerce applications

## Analiza porównawcza nierelacyjnych baz danych w zastosowaniach e-commerce

Grzegorz Rożek\*, Kacper Saweczko\*, Małgorzata Plechawska-Wójcik

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

### Abstract

In this article, a comparative analysis of non-relational databases was conducted to determine the best database for e-commerce systems. The research thesis was "MongoDB is the best choice for e-commerce applications. Non-relational systems such as MongoDB and Apache Cassandra were used for the study and the results were compared with a relational PostgreSQL database. The main research criterion was performance testing of several types of queries based on execution time. To implement the research, typical e-commerce databases were created and then tested in a .NET test application created by authors. In addition, the difference in community support between non-relational and relational systems was determined. The research showed that MongoDB is best suited for e-commerce systems.

*Keywords:* non-relational database; MongoDB; Cassandra; e-commerce

### Streszczenie

W tym artykule przeprowadzono analizę porównawczą nierelacyjnych baz danych w celu wyłonienia najlepszej bazy do systemów e-commerce. Postawiono tezę badawczą „MongoDB jest najlepszym wyborem w zastosowaniach e-commerce. Do badań wykorzystano systemy nierelacyjne takie jak MongoDB oraz Apache Cassandra, a wyniki zestawiono z relacyjną bazą PostgreSQL. Głównym kryterium badawczym były testy wydajności kilku rodzajów zapytań na podstawie czasu realizacji. Do realizacji badań zostały stworzone typowe dla e-commerce bazy danych, które następnie poddano testom w stworzonej aplikacji testowej w środowisku .NET. Dodatkowo określono różnicę we wsparciu społeczności pomiędzy systemami nierelacyjnymi a relacyjnymi. Badania wykazały, że do systemów e-commerce najlepiej przystosowane jest MongoDB.

*Słowa kluczowe:* nierelacyjna baza danych; MongoDB; Cassandra; handel elektroniczny

\*Corresponding authors

Email address: [grzegorz.rozek@pollub.edu.pl](mailto:grzegorz.rozek@pollub.edu.pl), [kacper.saweczko@pollub.edu.pl](mailto:kacper.saweczko@pollub.edu.pl)

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

## 1. Wstęp

Handel elektroniczny [1], nazywany także e-handlem albo e-commerce, odgrywa coraz większą rolę w dzisiejszym globalnym systemie gospodarczym. Obejmuje on sprzedaż i zakup produktów oraz usług za pośrednictwem Internetu i innych elektronicznych środków komunikacji. Rozwój technologii i wzrost popularności Internetu przyczyniły się do dynamicznego rozwoju handlu elektronicznego na całym świecie. E-commerce oferuje wiele korzyści zarówno dla konsumentów, jak i dla przedsiębiorstw. Klienci mogą łatwo porównywać produkty i ceny, robić zakupy o dowolnej porze oraz mieć dostęp do szerokiego wyboru towarów z całego świata. Umożliwia on również wygodne płatności online, co eliminuje konieczność fizycznego udawania się do sklepu i płacenia gotówką. Klienci mają do dyspozycji różne opcje dostawy, w tym dostawy do domu lub odbioru w punkcie odbioru. Coraz większa popularność internetowych serwisów e-commerce wymaga wybrania odpowiedniej bazy danych, która zapewni skalowalność oraz bezpieczny i szybki dostęp do danych dla cały czas zwiększającej się liczby klientów.

Celem pracy jest przeprowadzenie analizy porównawczej nierelacyjnych baz danych w zastosowaniach

e-commerce oraz porównanie wyników z wybraną bazą relacyjną. Analiza skupia się na porównaniu architektur omawianych baz oraz przetestowaniu ich wydajności. Pomoże to wskazać, który z wybranych systemów nierelacyjnych baz danych jest najbardziej odpowiedni dla systemów e-commerce. Dodatkowo wyniki analizy zostaną zestawione z bazą relacyjną, aby porównać różnice w ich działaniu przy liczbie sięgającej 1000 jednoczesnych użytkowników. W artykule postawiono tezę badawczą „MongoDB jest najlepszym wyborem w zastosowaniach e-commerce.” Dodatkowo zweryfikowane zostaną następujące pomocnicze pytania badawcze:

- Czy baza MongoDB jest bardziej przystosowana do zastosowań e-commerce niż Cassandra?
- Czy nierelacyjne bazy danych są wydajniejsze niż relacyjne w systemach e-commerce?
- Czy wybrane bazy nierelacyjne mają większe wsparcie społeczności niż wybrany system relacyjny?

## 2. Przegląd literatury

W wielu artykułach poruszona została kwestia wydajności i zastosowań różnych baz danych.

W artykule [2] przedstawiona została struktura i składniki baz danych dla systemów e-commerce w świecie rzeczywistym. Opisano szczegółowy łańcuch wartości e-commerce oraz wymagania dotyczące danych. Na potrzeby artykułu przeanalizowano łańcuch składający się z ośmiu procesów biznesowych. Następnie przedstawiono schemat bazy danych dla e-commerce w notacji UML. Analiza projektu skupiła się na podstawach sprawnego działania systemów przetwarzania transakcji e-commerce. Wynika z niego, że z punktu widzenia projektowania baz danych interesującym problemem badawczym jest to, jakie struktury baz danych są potrzebne, aby najefektywniej wspierać dostosowanie i personalizację. Na przykład, jak i jakie dane trzeba przechwytać, aby zbudować hurtownię internetową do personalizacji, a następnie, jak komunikować się z użytkownikami systemów.

W badaniu [3], aby zbudować udany i skalowalny sklep internetowy użyto dokumentowej bazy danych do włączenia przepływu pracy e-commerce. Projekt i strategia pracy e-commerce została oparta o MongoDB, która jest szeroko stosowaną bazą danych NoSQL do przetwarzania aplikacji biznesowych na dużą skalę. Autorzy zaczęli od analizy wydajności MongoDB, następnie opracowali schemat bazy danych dla systemu e-commerce. Przeanalizowali oni także elementy, które są najbardziej istotne podczas wymiany informacji pomiędzy systemem a klientem. Zorientowana na dokumenty baza danych NoSQL, taka jak MongoDB, ma wiele zalet w porównaniu z relacyjnymi bazami danych, np. może zawierać tysiące rekordów danych z odpowiednim elastycznym modelem. Również tego rodzaju modele danych, takie jak modele projektowania schematów, są przydatne do dalszego rozwoju nowych systemów przyjętych zasobów.

W pracy [4] autorzy opracowali badanie porównawcze, w którym oceniają wydajność dwóch bardzo rozpowszechnionych w tej dziedzinie baz danych: MySQL jako relacyjnej bazy danych oraz MongoDB jako bazy NoSQL. Do realizacji tej konfrontacji autorzy wykorzystali Yahoo! CloudServing Benchmark - YCSB. Miało to na celu udzielenie niezbędnej pomocy i wsparcia zainteresowanym podmiotom z branży Big Data i Cloud Computing w podjęciu decyzji o wyborze najlepszego rozwiązania bazodanowego, jakie powinno zostać przyjęte w ich firmach. Po analizie wyników eksperymentalnych autorzy doszli do wniosku, że wybór przyjmowanego rozwiązania zależy od zestawu parametrów takich jak wielkość środowiska, charakter i szacowana częstotliwość wykonywanych operacji.

W artykule [5] sprawdzono piętnaście kategorii baz danych NoSQL, aby poznać cechy każdej z nich. Zaproponowano pewne zasady i przykłady, które pozwolą wybrać odpowiednią bazę danych NoSQL dla różnych branż. Autorzy opisali cechy baz danych NoSQL. Na tej podstawie starali się wybrać najbardziej efektywny dla każdego rodzaju system. Podsumowując, jeśli firma porzuca relacyjną bazę danych i przechodzi na bazę NoSQL, musi rozważyć cechy danych firmy, aby znaleźć odpowiednią bazę danych. Dane transakcyjne bran-

ży e-commerce często muszą być powiązane, odpowiednią kategorią NoSQL jest szeroka rodzina kolumn, a Apache HBase jest dobrym wyborem.

Przeprowadzone badania w artykule [6] skupiły się na identyfikacji najbardziej wydajnego systemu przechowywania danych pod względem czasu odpowiedzi, porównując dwa najbardziej reprezentatywne systemy bazodanowe z dwóch kategorii (NoSQL i relacyjnych), tj. MongoDB oraz PostgreSQL. Ocena oparta jest na rzeczywistych, biznesowych scenariuszach i ich kolejnych zapytaniach jak również infrastruktur bazowych i kończy się na potwierdzeniu wyższości PostgreSQL. W szczególności baza ta jest czterokrotnie szybsza pod względem czasu odpowiedzi w większości przypadków. Zaobserwowano również, że średni czas odpowiedzi jest zmniejszony o połowę przy użyciu indeksów niemal we wszystkich przypadkach, podczas gdy redukcja jest znacznie mniejsza w przypadku PostgreSQL.

W badaniu [7] przedstawiono przypadek, który opierał się na dwóch stronach internetowych e-commerce. W trakcie badania połączono dwie różne bazy danych przy użyciu usług internetowych do pobierania danych. Wszystkie komponenty scalające były przechowane na usługach chmurowych. Każdy proces wsadowy w systemie synchronizacji odbywał się online i automatycznie wykorzystywał narzędzie cron. W opisywanym badaniu przedstawiono wyniki testów integracji danych na usługach webowych oraz pomiar czasu wykonania systemu synchronizacji. Po przetestowaniu dwóch użytych serwisów internetowych okazało się, że wszystkie przeszły test integracji danych. Następnie dokonano pomiarów na systemie synchronizacji patrząc na wyświetlany czas wykonania. Wykorzystano od 100 do 1000 danych i różne warunki. Uzyskano czas wykonania od 4 do 5 sekund dla każdej danej w pomiarze. Wyniki badań wskazały, że integracja danych z dwóch różnych baz danych z wykorzystaniem usług internetowych i systemów synchronizacji jest znośna i odpowiednia dla ilości danych poniżej 10000 rekordów przy stanie bazy już wypełnionej.

W artykule [8] przedstawiono taksonomię dotyczącą systemów NoSQL. Używając tej perspektywy, dokonano porównań różnych systemów NoSQL używając wielu aspektów, w tym architektury systemu, modelu danych, języka zapytań, API klienta, skalowalności i dostępności. Pogrupowano obecne systemy NoSQL ze względu na ich model danych: klucz-wartość, rodzina kolumn, dokument, graf, natywny XML. Opisano również scenariusze zastosowań dla każdej kategorii, aby pomóc w wyborze odpowiedniego systemu NoSQL dla danej aplikacji.

W pracy [9] autorzy zaproponowali koncepcję systemu e-commerce, który w procesie tworzenia rekomendacji, gromadzi i wykorzystuje dane pochodzące z profili społecznościowych swoich użytkowników. Takie podejście do modelowania architektury zostało opracowane w ramach projektu aplikacji internetowej typu mashup, która integruje się z API Facebooka. Opisane zostało, jakie dane można pozyskać z Facebooka, zaproponowano sposób ich przechowywania oraz wska-

zono, w jaki sposób informacje z profilu użytkownika mogą poprawić efektywność systemu rekomendacji handlu elektronicznego.

W badaniach z artykułu [10] omówiono różne aspekty MongoDB. Przedstawiono zalety korzystania z systemu, a następnie dokonano zestawienia najpopularniejszych zagadnień wykorzystujących tę bazę danych. Przeanalizowano główne problemy systemu takie jak konstrukcja schematu bazy danych czy też brak niezawodności. Wyniki tego badania otworzyły nowe drogi dla przyszłych badań nad wydajnością dostępu do danych kiedy występują hotspoty w danych czyli sektory, w których występuje wiele żądań dostępu w określonym momencie. Założono, że dostęp do wszystkich danych będzie uzyskiwany przy użyciu tych samych wzorców.

W artykule [11] dokonano porównania cech wyróżniających narzędzia relacyjnych i nierelacyjnych bazy danych. Zestawiono kluczowe aspekty największych relacyjnych systemów baz danych takich jak Oracle i MySQL. W kwestii nierelacyjnych systemów dokonano klasyfikacji na podstawie sposobu organizacji danych. Czynniki podlegającymi porównaniu były np.: przepustowość, skalowalność, łatwości manipulacji danymi czy duplikacja danych.

Alternatywne wykorzystanie nierelacyjnej bazy Cassandra przedstawiono w [12]. Badania skupiły się na utrzymaniu i magazynowaniu danych strumieniowych z różnych platform.

W artykule [13] przedstawiono zintegrowany 8-procesowy łańcuch wartości potrzebny systemowi e-commerce i związane z nim dane na każdym etapie łańcucha wartości.

Przegląd literatury okazał się pomocny do wyboru metod badawczych, sposobu przeprowadzenia badań i wizualizacji wyników.

### 3. Wykorzystane systemy bazodanowe

#### 3.1. Baza MongoDB

Baza MongoDB [14] jest to stworzony w języku C++ magazyn danych NoSQL. Główną strukturą danych są kolekcje, które zawierają dokumenty. Dokumenty posiadają unikalny klucz specjalny o nazwie `_id`, który najczęściej jest typu "ObjectId", a którego używa się do jednoznacznej identyfikacji dokumentów. Dokumenty MongoDB poddawane są serializacji jako obiekty JSON i przechowywane wewnętrznie przy użyciu binarnego kodowania BSON. Najpopularniejszymi formatami plików, których styl przyjmują dokumenty to: XML, JSON, YAML i CSV. Do składowania danych wykorzystuje się zagnieżdżanie dokumentów lub referencje do innych dokumentów na żądanie użytkownika. System ten jest aktualnie najpopularniejszym systemem wybieranym do rozwiązań NoSQL, ponieważ charakteryzuje się dużą skalowalnością i elastycznością w zakresie przechowywania i przetwarzania danych. W chwili obecnej bazę MongoDB najczęściej uruchamia się w chmurze obliczeniowej ale do dyspozycji użytkowników są również narzędzia takie jak aplikacja kliencka Compass.

#### 3.2. Baza Apache Cassandra.

Apache Cassandra [15] jest to produkt open source stworzony przez firmę Facebook. Podobnie jak MongoDB jest to rozwiązanie NoSQL, jednak Cassandre wyróżnia to, że jest rozproszona i przeznaczona do pracy na dużej ilości serwerów. Cassandra jest kolumnową bazą danych, zaprojektowaną do gromadzenia dużej ilości danych. Posiada odpowiednie narzędzia do utrzymania klasy niezawodności takie jak silnik BigTable oraz Dynamo wspomagające przechowanie i replikację. Główne zalety tej bazy danych to: pełna replikacja bazy danych, niskie opóźnienia przy wysokiej dostępności, duża elastyczność schematu czy rozwój klastra. Do wykonywania zapytań wykorzystywany jest wewnętrzny język CQL (Cassandra Query Language). Do wad tego rozwiązania można zaliczyć takie aspekty systemu jak: problemy z zarządzaniem pamięcią JVM oraz brak obsługi agregatów. Organizacja danych opiera się o klastry węzłów.

#### 3.3. Baza PostgreSQL

Baza PostgreSQL to jedyna relacyjna baza w zestawieniu systemów bazodanowych wybranych przez autorów pracy. Jest to system obiektowo-relacyjny będący produktem open source [16]. PostgreSQL zapewnia wysoką ochronę integralności danych i odporność na błędy. Podobnie jak w większości relacyjnych baz danych, podstawową strukturą do przechowywania danych jest tabela. Do manipulacji danymi służy język SQL, jednak można również używać funkcji i procedur w języku PL/pgSQL. Integralność danych zapewniają klucze główne i obce definiowane na poziomie tabeli. Dostępne są trzy rodzaje replikacji danych: synchroniczna, asynchroniczna i logiczna. Baza posiada dobrze rozwinięty system kontroli dostępu, szyfrowanie i uwierzytelnianie wieloczynnikowe. Na chwilę obecną najbardziej znane społeczności wady PostgreSQL to porównywalnie niska prędkość odczytu oraz brak dostępności na wszystkich hostach.

### 4. Plan badań

Do badań utworzone zostały trzy niezależne bazy danych w wybranych systemach. W celu weryfikacji postawionej tezy i pytań badawczych stworzona została aplikacja testowa w środowisku .NET, która symuluje działanie systemu e-commerce z wykorzystaniem stworzonych baz danych. Aplikacja pozwala na rejestrację użytkowników oraz zarządzanie produktami i zamówieniami klientów. Zastosowany został wzorzec projektowy repozytorium, aby w zależności od potrzeb wykorzystać właściwą bazę danych. W przypadku bazy PostgreSQL oprócz automatycznie indeksowanych kluczy głównych dodany został unikalny indeks na kolumnie email w tabeli użytkownika. Natomiast dla bazy Cassandra tabele zostały zaprojektowane biorąc pod uwagę zapytania wymagane przez aplikację. Oznacza to, że jeżeli potrzebne były dane według id oraz nazwy, utworzono dwie tabele powielające dane. W pierwszej tabeli kluczem partycji było pole id, a w drugiej pole nazwa. Zastosowany został także indeks pomocniczy dla tabeli

zamówień na kolumnie ze statusem. Do wykonywania operacji na bazie bezpośrednio z aplikacji wykorzystano wspierane przez twórców biblioteki: Entity Framework Core, DataStax C# Driver i MongoDB C# Driver. W operacjach korzystano ze składni zapytań LINQ [17], która jest tłumaczona na odpowiednie zapytania na bazie, przez wykorzystane biblioteki. Za pomocą biblioteki BenchmarkDotNet [18] przeprowadzone zostały następujące testy wydajności systemów nierelacyjnych dla następujących operacji występujących w systemie e-commerce:

- dodawanie użytkownika, produktu i zamówienia,
- wyszukiwanie użytkowników, zamówień i produktów,
- usuwanie użytkownika, produktu i zamówienia,
- aktualizacja produktu i zamówienia.

Dla zapewnienia wiarygodności testów zostały one powtórzone 1000 razy dla każdego systemu bazodanowego.

Do przedstawienia poziomu wsparcia społeczności zostały wykorzystane platformy GitHub i Stack Overflow. Technika pozwalająca określić zainteresowanie daną tematyką jest analiza popularności polegająca na zliczeniu rozpoczętych dyskusji na temat systemów bazodanowych na forum Stack Overflow oraz liczby gwiazdek repozytorium na GitHub.

#### 4.1. Struktury badanych danych

Struktura danych opiera się na trzech głównych encjach: użytkownika, produktu i zamówienia. W przypadku bazy PostgreSQL, aby nie duplikować danych zastosowano normalizację. Powstały w ten sposób następujące tabele: adres, który jest w relacji z użytkownikiem i zamówieniem, kategoria produktu związana z tabelą produktów oraz szczegóły płatności, metoda płatności, status płatności, status zamówienia, sposób dostawy, będące w relacji z tabelą zamówienia. W przypadku użytych baz NoSQL zastosowano denormalizację. Wymagane dane umieszczono bezpośrednio w głównych encjach.

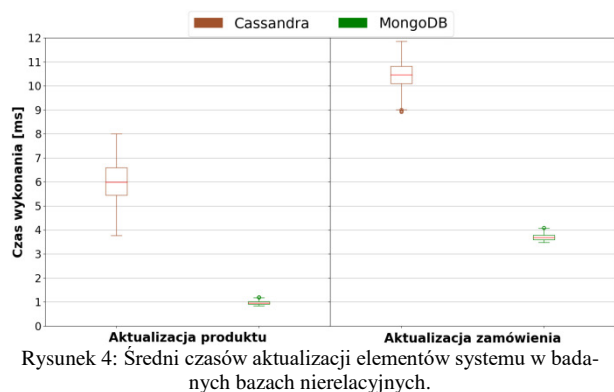
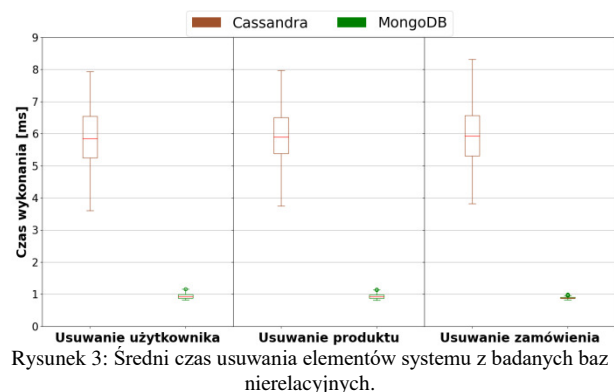
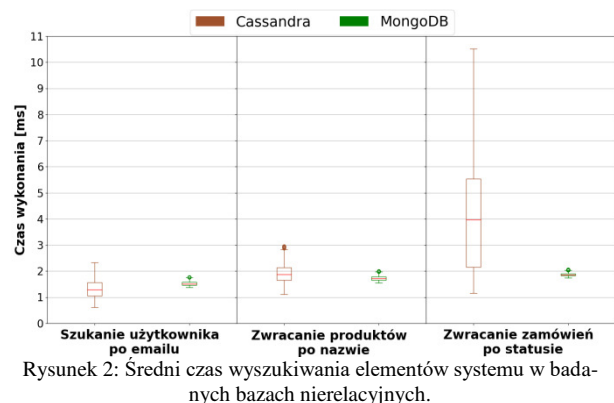
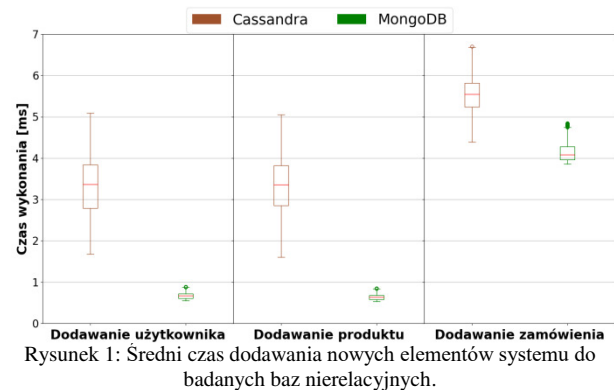
Operacja zwracania zamówień zawiera również informacje o powiązanych z nimi produktami. Ponadto usunięcie użytkownika lub produktu nie wpływa na rekordy w tabeli z zamówieniami.

### 5. Wyniki badań

#### 5.1. Analiza wyników wydajności

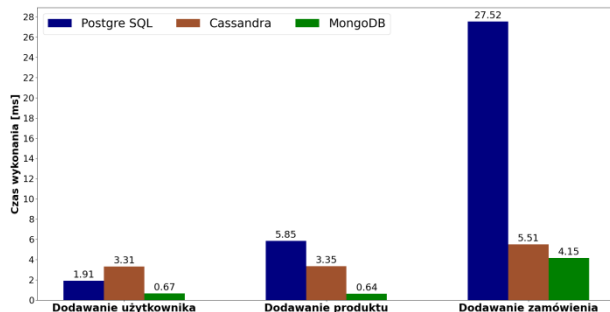
Na Rysunkach 1-4 przedstawiono wykresy pudełkowe ilustrujące różnice w czasach wykonywania operacji na bazach nierelacyjnych za pośrednictwem aplikacji testowej. W większości przypadków występuje sytuacja, gdzie MongoDB okazuje się szybsza od Cassandra. Różnice średnich czasów wynoszą od około milisekundy do niecałych dwóch milisekund w przypadku operacji dodawania. Największa nieregularność występuje przy zapytaniach wyszukiwujących. Na uwagę zasługuje operacja zwracania produktów po nazwie gdzie Cassandra okazała się lepsza o 0.17 ms. Warto również nadmienić, że największa różnica w czasie występuje podczas operacji zwracania wszystkich zamówień i wynosi około 22 ms. W przypadku operacji usuwania Mon-

goDB jest o około 5 ms szybsze. Podczas aktualizacji Cassandra, wypada około 5-6 ms gorzej niż MongoDB.

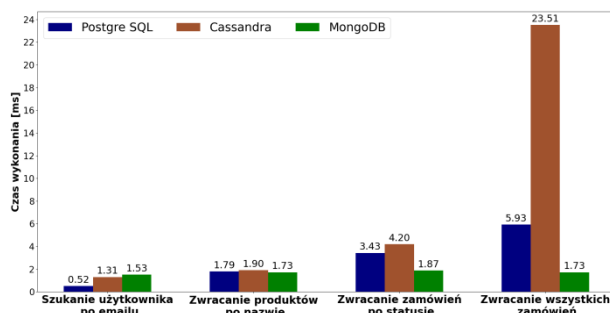


Na Rysunkach 4-8 zobrazowano zestawienie średnich czasów wszystkich wybranych systemów bazodanowych. Baza PostgreSQL nie wypada lepiej niż MongoDB za wyjątkiem operacji wyszukania użytkownika

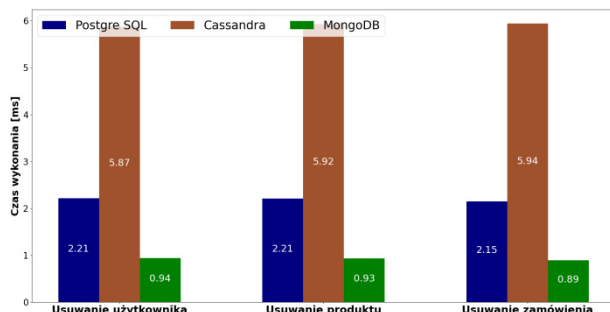
przy pomocy adresu email. Natomiast w większej liczbie testów system ten okazał się być szybszy niż Cassandra. Największe różnice w średnim czasie między systemami nierelacyjnymi a PostgreSQL występują w przypadku operacji na zamówieniach takich jak dodawanie i aktualizacja.



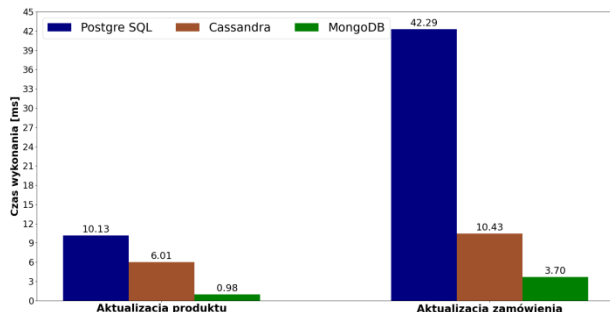
Rysunek 5: Średni czas trwania dodawania elementów w poszczególnych bazach danych.



Rysunek 6: Średni czas zwracania elementów w poszczególnych bazach danych.



Rysunek 7: Średni czas trwania usuwania elementów w poszczególnych bazach danych.



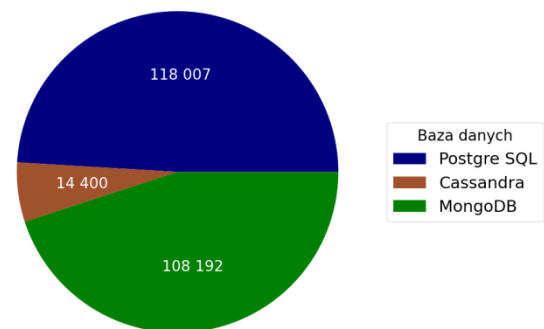
Rysunek 8: Średni czas trwania aktualizacji elementów w poszczególnych bazach danych.

## 5.2. Analiza wsparcia społeczności

Na forum Stack Overflow [19], które jest największą społecznością przeznaczoną do dzielenia się wiedzą

przez deweloperów, sprawdzono dla każdej badanej bazy liczbę pytań, które posiadają odpowiedź od użytkowników na temat danej technologii. Na Rysunku 9 można zauważyć, że najwięcej pytań, na które udzielono odpowiedzi posiada relacyjna baza PostgreSQL – ok. 118 tys. Niewiele mniej jest w przypadku MongoDB – ok. 108 tys. Natomiast zdecydowanie najmniej posiada Cassandra: ok. 14 tys.

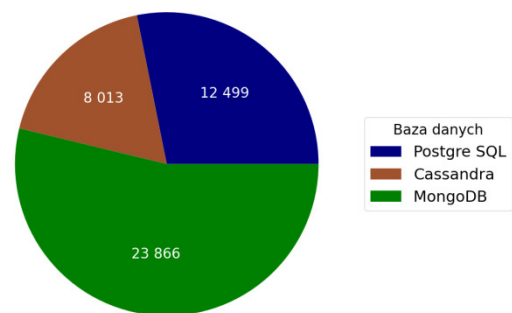
Liczba pytań z odpowiedziami na portalu Stack Overflow



Rysunek 9: Zestawienie liczby pytań z odpowiedziami na Stack Overflow wybranych systemów bazodanowych na dzień 08.06.2023r.

Z kolei w serwisie GitHub [20] sprawdzono posiadaną liczbę gwiazdek przez oficjalne repozytoria. Stanowią one wyraz uznania społeczności i wpływają na ranking repozytorium na stronie. Na przedstawionym Rysunku 10 zdecydowanym faworytem jest baza MongoDB, która posiada ok. 24 tys. gwiazdek. Drugie miejsce zajmuje PostgreSQL, które posiada ich ok. 12 tys. Na ostatnim miejscu znajduje się Cassandra z 8 tys. gwiazdek.

Liczba posiadanych gwiazdek repozytorium na platformie GitHub



Rysunek 10: Zestawienie liczby gwiazdek repozytorium na platformie GitHub dla wybranych systemów bazodanowych na dzień 08.06.2023r.

## 6. Wnioski

Wykresy na rysunkach 1-8 wizualizujące przeprowadzone badania jednoznacznie potwierdzają postawioną tezę. MongoDB jest najlepiej przystosowaną bazą danych do systemów e-commerce. Baza ta jest najszybsza w 11 na 12 badanych operacjach. Większość wybranych do badań operacji wykonuje średnio poniżej 1 ms, a najwolniej pojedynczą operację wykonuje nieco powyżej 4ms. To doskonały wynik przy 1000 prób.

Różnica w wydajności pomiędzy systemami relacyjnymi a nierelacyjnymi najbardziej zauważalna jest



w przypadku operacji aktualizujących. W zależności od operacji można stwierdzić, że PostgreSQL cechuje podobna wydajność do Cassandra. Natomiast PostgreSQL okazało się lepsze niż MongoDB tylko w jednej z 12 badanych operacji. Społeczność udzielająca się na wybranych platformach prezentuje porównywalną aktywność w przypadku wybranych do badań baz nierelacyjnych i relacyjnych. Dzięki zebranych danym można stwierdzić, że wsparcie jest na bardzo podobnym poziomie. Warto nadmienić jednak, że bazy nierelacyjne są stosunkowo młode w porównaniu do relacyjnych. Porównywalny poziom dyskusji na najpopularniejszym forum świadczy o wysokiej dynamice rozwoju tych rozwiązań.

Badania można kontynuować pod kątem bezpieczeństwa wybranych baz danych w systemach e-commerce. Bezpieczeństwo danych jest ważnym aspektem przy projektowaniu systemów więc może to być ciekawy kierunek. Można również porównać skalowalność przy jednoczesnym korzystaniu przez wielu użytkowników.

### Literatura

- [1] A. Molla, P.S. Licker, E-commerce systems success: An attempt to extend and respecify the DeLone and MacLean model of IS success, *Journal of Electronic Commerce Research* 2 (2001) 131-141.
- [2] I. Y. Song, K. Y. Whang, Database design for real-world e-commerce systems, *IEEE Data Eng. Bull.* 23.1 (2000) 23-28.
- [3] D. Ramesh, E. Khosla, S. N. Bhukya, Inclusion of e-commerce workflow with NoSQL DBMS: MongoDB document store, *IEEE international conference on computational intelligence and computing research (ICCIIC)* (2016) 1-5.
- [4] H. Matallah, G. Belalem, K. Bouamrane, Comparative study between the MySQL relational database and the MongoDB NoSQL database, *International Journal of Software Science and Computational Intelligence (IJSSCI)* 13(3) (2021) 38-63.
- [5] J. K. Chen, W. Z. Lee, An Introduction of NoSQL Databases based on their categories and application industries, *Algorithms* 12(5) (2019) 1-16, <https://doi.org/10.3390/a12050106>.
- [6] A. Makris, K. Tserpes, G. Spiliopoulos, D. Anagnostopoulos, Performance Evaluation of MongoDB and PostgreSQL for spatio-temporal Data, *EDBT/ICDT Workshops 2019 Joint Conference Workshops*, Lisbon, Portugal (2019) 1-8.
- [7] A. J. Maulidin, F. Renaldi, F. R. Umbara, Online Integration of SQL and No-SQL Databases using RestAPIs: A Case on 2 furniture e-Commerce Sites, *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)* (2020) 261-266.
- [8] V. N. Gudivada, S. Jothilakshmi, D. Rao, Data management issues in big data applications, *ALLDATA* 15 (2015) 16-21.
- [9] D. Fijałkowski, R. Zatoka, An architecture of a Web recommender system using social network user profiles for e-commerce, *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)* (2011) 287-290.
- [10] A. Chauhan, A Review on Various Aspects of MongoDB Databases, *International Journal of Engineering Research & Technology (IJERT)* 8(5) (2019) 90-92.
- [11] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, D. Gosain, A survey and comparison of relational and non-relational database, *International Journal of Engineering Research & Technology* 1(6) (2012) 1-5.
- [12] A. Kumar, G. Vijaya. Streaming data analysis using apache cassandra and zeppelin. *IJISSET-International Journal of Innovative Science, Engineering & Technology* 3 (2016) 8-15.
- [13] SONG, Il-Yeol; WHANG, Kyu-Young. Database design for real-world e-commerce systems. *IEEE Data Eng. Bull.* 23.1 (2000) 23-28.
- [14] What is MongoDB - MongoDB documentation, <https://www.mongodb.com/docs/manual>, [08.06.2023].
- [15] Apache Cassandra Documentation – overview, <http://cassandra.apache.org/doc/latest/cassandra/architecture/overview.html>, [08.06.2023].
- [16] PostgreSQL - About, <https://www.postgresql.org/about>, [08.06.2023].
- [17] LINQ – overview <https://learn.microsoft.com/en-us/dotnet/csharp/linq/> [26.06.2023].
- [18] BenchmarkDotNet overview, <https://benchmarkdotnet.org/>, [26.06.2023].
- [19] StackOverflow – forum programistyczne, <https://stackoverflow.com>, [08.06.2023].
- [20] GitHub – serwis internetowy, <https://github.com>, [08.06.2023].