POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA (INF)

SPECJALNOŚĆ: SYSTEMY I SIECI KOMPUTEROWE (ISK)

PRACA DYPLOMOWA

MAGISTERSKA

Analiza porównawcza relacyjnych i nierelacyjnych baz danych na wybranych przykładach

Comparative analysis of relational and   
non-relational databases on selected examples

AUTOR:

Paweł Idziak

PROWADZĄCY PRACĘ:

Dr inż. Jacek Cichosz, W4\K2

OCENA PRACY:

WROCŁAW, 2019

**Spis treści**

[Spis rysunków 4](#_Toc5374889)

[Spis tabel 5](#_Toc5374890)

[Spis listingów 6](#_Toc5374891)

[Skróty 7](#_Toc5374892)

[1. Wstęp 8](#_Toc5374893)

[1.1. Rozwój baz danych 8](#_Toc5374894)

[1.2. Cel pracy 8](#_Toc5374895)

[1.3. Zastosowanie baz danych 9](#_Toc5374896)

[1.4. Uzasadnienie wyboru baz danych 13](#_Toc5374897)

[1.5. Przedstawienie struktury pracy 14](#_Toc5374898)

[2. Analiza wybranych baz danych 15](#_Toc5374899)

[2.1. MySQL 16](#_Toc5374900)

[2.2. PostgreSQL 18](#_Toc5374901)

[2.3. MongoDB 19](#_Toc5374902)

[3. Użyte technologie 20](#_Toc5374903)

[3.1. NodeJS 20](#_Toc5374904)

[3.2. ExpressJS 20](#_Toc5374905)

[3.3. Angular 7 20](#_Toc5374906)

[3.4. TypeScript 20](#_Toc5374907)

[3.5. GIT 20](#_Toc5374908)

[4. Implementacja 22](#_Toc5374909)

[4.1. Założenia projektowe 22](#_Toc5374910)

[4.2. Wykorzystane technologie i narzędzia 22](#_Toc5374911)

[4.3. Opis działania systemu 22](#_Toc5374912)

[5. Analiza wydajnościowa 23](#_Toc5374913)

[6. Podsumowania 24](#_Toc5374914)

[Literatura 25](#_Toc5374915)

[Dodatek A 26](#_Toc5374916)

# Spis rysunków

[Rys. 1. Tendencja popularności baz danych - marzec 2019. Źródło: https://db-engines.com/en/ranking\_trend [Udostępniono: 28 marzec 2019] 14](#_Toc5374917)

[Rys. 2. Logo MySQL - www.mysql.com. [03 marzec 2019] 16](file:///D:\Workspace\mgr\SQL-NoSQL-comparison\mgr_praca.docx#_Toc5374918)

[Rys. 3 Przykładowy diagram ERD 17](#_Toc5374919)

# Spis tabel

[Tab. 1. Przykład podpisu tabeli 20](#_Toc5374920)

# Spis listingów

[Listing. 1. Początkowe żądanie HTTP 21](#_Toc5374921)

# Skróty

**OGC** (ang. *Open Geospatial Consortium*)

**XML** (ang. *eXtensible Markup Language*)

**SOAP** (ang. *Simple Object Access Protocol*)

**WSDL** (ang. *Web Services Description Language*)

**UDDI** (ang. *Universal Description Discovery and Integration*)

**GIS** (ang. *Geographical Information System*)

**SDI** (ang. *Spatial Data Infrastructure*)

**ISO** (ang. *International Standards Organization*)

**WMS** (ang. *Web Map Service*)

**WFS** (ang. *Web Feature Service*)

**WPS** (ang. *Web Processing Service*)

**GML** (ang. *Geography Markup Language*)

**SRG** (ang. *Seeded Region Growing*)

**SOA** (ang. *Service Oriented Architecture*)

**IT** (ang. *Information Technology*)

1. Wstęp

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus. Aliquam erat volutpat. Integer maximus est turpis, ut bibendum ligula accumsan et. Ut eget vestibulum libero. Aliquam erat volutpat. Nullam placerat mauris a lectus tincidunt, et aliquet turpis aliquam. Etiam in malesuada lacus. Proin dignissim augue sit amet auctor elementum. Suspendisse potenti. Vivamus suscipit vulputate massa ac molestie. Suspendisse a justo porttitor, commodo mi at, placerat risus. Integer lobortis augue ac neque suscipit, vel sodales lacus fringilla.

WSTEP – co jest w pracy, co jest celem i jej zakres ale bez wchodzenia w szczególy.

* To, że w dzisiejszych czasach obserwuje sie gwałtowny wzrost gromadzonych danych (pokazad statystki z cisco)
* To, że wzrost danych wiąże się z zarządzaniem tymi danymi w odpowiedni sposób.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. Rozwój baz danych
  2. Cel pracy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. Zastosowanie baz danych

Coś o zastosowaniu, później definicja, DBMS, architektura, rodzaje

Internetowy słownik Merriam–Webster [1] definiuję bazę danych jako „dużą kolekcję danych, zorganizowaną w celu szybkiego przeszukiwania i dostępu do informacji”. Rozwijając tą definicję, baza danych jest zorganizowanym zbiorem danych, zapisanym z określonymi regułami oraz przechowywanym w określony sposób. W znaczeniu technologii komputerowych, definicja bazy danych obejmuje dane cyfrowe gromadzone według ogólnych zasad przyjętych dla danego systemu, przeznaczonego do wykonywania określonych operacji na danych.

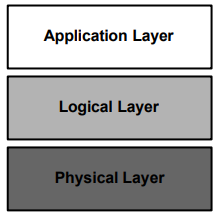
Zarządzanie bazą danych realizowane jest poprzez *DBMS* [2](ang. *Database Management System*), czyli *system zarządzania bazą danych*. Jest to najczęściej oprogramowanie, składające się z zestawu bibliotek oraz narzędzi, które umożliwiają użytkownikowi między innymi zdefiniowanie oraz utworzenie bazy. DBMS posiada interfejs dostępu do danych oraz funkcjonalności służące do zarządzania, analizowania czy przechwytywania danych. Głównym założeniem DBMS jest współdziałanie z użytkownikami końcowymi, aplikacjami, a samą bazą danych. Funkcjonalności, jakie spełnia system zarządzania bazą danych obejmuje:

* **Tworzenie.** Każda baza danych może być tworzona na różne sposoby. DBMS posiada określoną procedurę tworzenia bazy danych, dzięki czemu użytkownik nie ma potrzeby interakcji z niskopoziomowymi strukturami plików. System zarządzania bazą udostępnia dostarcza odpowiedni dostęp potrzebny użytkownikom poprzez zdefiniowany interfejs.
* **Selekcja i uaktualnienie.** Systemy DBMS powiadają funkcjonalności tworzenia zapytań o dane spełniające określone kryteria (np. wszystkie zamówienia pojedynczego użytkownika), oraz zapytań realizujących funkcje usuwania czy aktualizacji określonych danych.
* **Wielozadaniowość.** W sytuacji, gdy kilka procesów (np. aplikacji) korzysta z bazy danych jednocześnie, system zapewnia dla każdego z połączenia dostęp do zasobów / funkcjonalności bazy. Wielozadaniowość zapewnia odebranie oraz przetworzenie każdego żądania bez negatywnego wpływu na inne elementy systemu.
* **Bezpieczeństwo.** System zarządzania bazą danych zapewnia mechanizm kontroli dostępu do poszczególnych elementów. Dzięki takiemu rozwiązaniu, operacje manipulowania danymi oraz ich strukturą są dozwolone jedynie dla uprawnionych użytkowników. System DBMS zawiera mechanizmy zarządzania użytkownikami, a zdefiniowanie odpowiedniej hierarchii użytkowników pozwala na dostęp do określonych zasobów dla danej grupy i / lub operacji wykonywanych na bazie danych.
* **Historia operacji.** Kolejną ważną funkcjonalnością DBMS jest tworzenie oraz zapisywanie historii operacji wykonanych na przestrzeni określonego czasu. Taką funkcjonalność można wykorzystać do śledzenia zmian, wykrywania błędów czy rekonstrukcji danych w sytuacji awarii systemu. Kolejną zaletą jest wykorzystanie funkcjonalności historii do tworzenia rejestru operacji oraz kopi zapasowej danych.

Bazy danych mają szerokie zastosowane w różnych dziedzinach. ………..

Cos o przeznaczeniach i przejście do architektury

Architekturę bazy danych można przedstawić za pomocą abstrakcyjnego standardu *ANSI-SPARC* (ang. *American National Standards Institute, Standards Planning And Requirements Committee*), na którym opiera się większość współczesnych baz danych. Schemat *ANSI-SPARC* został zdefiniowany w 1975 roku i przedstawia sposób projektowania systemów zarządzani bazami danych (*DBMS*). Koncepcja standardu opiera się na wyróżnieniu trzech głównych warstw systemu (ODNOŚNIK DO RYS), dlatego często spotykaną nazwą jest *Architektura trójwarstwowa*. Architektura trójwarstwowa posiada trzy główne komponenty, każdy rozumiany jako inny poziom abstrakcji.



Warstwa aplikacji (ang. *Application Layer*)

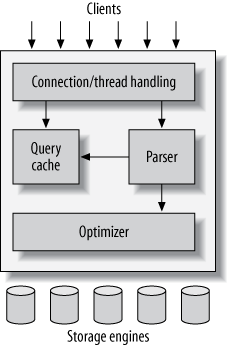
Warstwa aplikacji przedstawia pewien interfejs dostępny dla wszystkich użytkowników systemu, dzięki któremu użytkownicy mogą nawiązać interakcję z serwerem bazy danych. Wśród użytkowników można wyróżnić między innymi dwie główne grupy:

1. **Standardowi użytkownicy** – korzystają z systemu pośrednio, przez wcześniej zdefiniowany interfejs użytkownika.
2. **Zaawansowani użytkownicy** – współpracują z systemem bezpośrednio, bez użycia zdefiniowanej aplikacji. Użytkownicy tej grupy formułują żądania bezpośrednio za pomocą języka zapytań konkretnej bazy danych. Do tej grupy można zaliczyć administratorów lub programistów, którzy zajmują się implementacją aplikacji użytkowych.

Warstwa logiczna (ang. *Logical Layer*)

Warstwa logiczna zawiera podstawowe funkcjonalności zapewniające komunikację z fizyczną bazą danych. Warstwa logiczna przetwarza żądania użytkowników, definiuje reguły logiczne oraz wykonuje zdefiniowane obliczenia. Warstwa ta również pełni funkcję pośrednią pomiędzy otaczającymi ją warstwami (aplikacji oraz fizycznej) zapewniając przepływ danych. W tej części systemu istnieje wiele różnych implementacji komunikacji specyficznych dla danego systemu.

Warstwa fizyczna (ang. *Physical Layer*)



W zależności od stosowanej struktury organizacji danych w konkretnym systemie, można wyróżnić między innymi takie rodzaje baz danych jak:

* Relacyjne (RDBMS – ang. *Relational Database Management System*)
* Nierelacyjne (NoSQL – ang. *Non SQL / Non relational*)
* Obiektowe (OODBMS – ang. *Object-Oriented Database Management System*)
* Obiektowe – relacyjne (ORDBMS – ang. *Object-Relational Database Management System*)

Relacyjne bazy danych

Nierelacyjne bazy danych

Obiektowe bazy danych

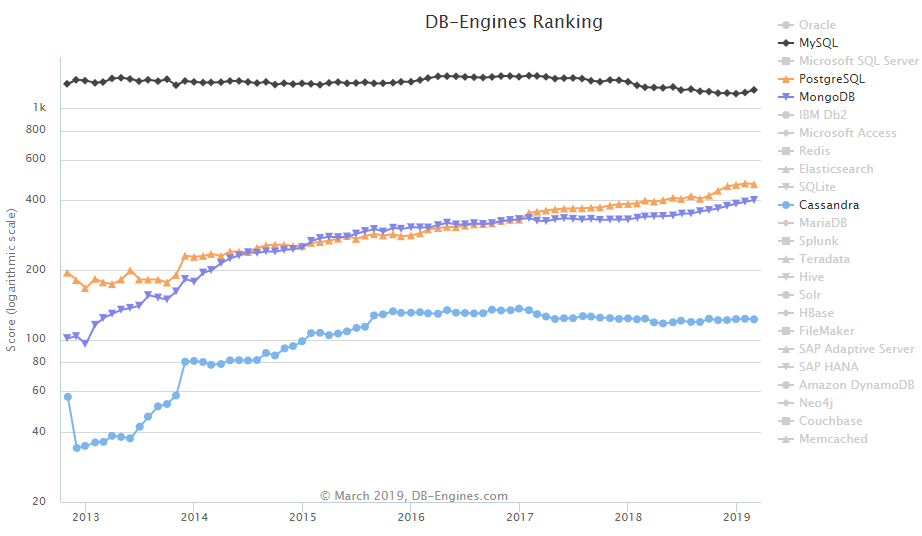
Obiektowo – relacyjne bazy danych

* 1. Uzasadnienie wyboru baz danych

W ramach niniejszej pracy magisterskiej zostały wybrane dwie relacyjne oraz dwie nierelacyjne bazy danych. Głównym kryterium wyboru konkretnych baz danych jest ich popularność datująca na marzec 2019r. Ranking popularności został zaczerpnięty ze strony internetowej *DB-ENGINES.COM* [3]. W momencie pisania pracy metoda użyta do zmierzenia popularności baz danych używa takich parametrów jak:

* **Liczba „wzmianek” o danym systemie na stronach internetowych**, mierzona liczbą wyników w popularnych internetowych wyszukiwarkach. Używane wyszukiwarki to Google, Bing oraz Yandex.
* **Ogólne zainteresowanie systemem bazy danych** mierzona jako częstotliwość wyszukiwania w Google Trends.
* **Częstotliwość dyskusji technicznych na temat systemu.** Dane są pobierane ze znanych witryn internetowych związanych z technologią IT takich jak Stack Overflow i DBA Stack Exchange.
* **Liczba ofert pracy, w których wspomniany jest dany system.** Oferty prace są przeglądane w serwisach takich jak Indeed oraz Simply Hired.

Jako relacyjne systemy zostały wybrane takie bazy jak *MySQL* oraz *PostgreSQL*, natomiast jako nierelacyjne *MongoDB* oraz *Cassandra*. Wykres poniżej przedstawia tendencję popularności wybranych systemów, na podstawie której zostały wybrane wyżej wymienione bazy.



Rys. 1. Tendencja popularności baz danych - marzec 2019. Źródło: <https://db-engines.com/en/ranking_trend> [Udostępniono: 28 marzec 2019]

* 1. Przedstawienie struktury pracy

Niniejsza praca magisterska została podzielona na …

1. Analiza wybranych baz danych

W tym rozdziale zostanie przedstawiona analiza wybranych relacyjnych i nierelacyjnych baz danych. Wybór baz danych został uzasadniony w rozdziale 1.4.

Jednym z analizowanych aspektów jest model bazy danych. Relacyjne bazy danych posiadają zdefiniowane modele, lecz nierelacyjne bazy danych nie posiadają spójnego modelu danych. Wśród nierelacyjnych modeli baz danych możemy wyróżnić między innymi TUTAJ JAKIŚ MODEL. Porównanie będzie zawierać przedstawienie cech wspólnych lub istotnych różnic.

MODEL DANYCH CZYLI JAK TE DANE SA PRZECHOWYWANE

Kolejnym analizowanym aspektem jest architektura bazy danych rozumiana jako elementy, z których składa się dany system oraz funkcjonalności takie jak między innymi NIE WIEM CZY TEN AKAPIT BĘDZIE OK. WYJDZIE W PRANIU.

Kolejnym analizowanym aspektem jest dostępność bazy danych. Każda baza danych potrzebuje pewnego sterownika (biblioteki), który służy do komunikacji pomiędzy aplikacją kliencką, a bazą danych. Aspekt dostępności uwzględnia takie założenia jak poziom trudności implementacji połączenia z bazą danych, rozumiany jako jakość instrukcji dostępnych w oficjalnej dokumentacji, oraz rodzaj języków programowania, poprzez które można ustanowić połączenie. Im większa liczba dostępnych języków programowania, tym mniejsze ograniczenie wyboru bazy danych.

Każdy aspekt analizy dla kolejnej bazy danych zostanie uwzględniony w poszczególnych podsumowaniach z uwzględnieniem cech nietypowych lub szczególnych dla danego rozwiązania.

* 1. MySQL



MySQL jest to system zarządzania relacyjnymi   
bazami danych (RDBMS – ang. *Relational Database   
Management System*), udostępniony na zasadzie   
*wolnego oprogramowania* (ang. *free software*) [4].   
Termin ten określa oprogramowanie, które może być uruchamiane, powielane, zmieniane oraz rozpowszechniane przez użytkowników. Więcej informacji na temat *wolnego oprogramowania* można znaleźć na oficjalnej stronie FSF (ang. *Free Software Foundation)* [4].

Rys. 2. Logo MySQL - [www.mysql.com](http://www.mysql.com). [03 marzec 2019]

Obecnie (marzec 2019) MySQL rozwijany jest przez firmę *Oracle*, lecz pierwsze wydanie systemu datowane jest na 23 maja 1995 roku. Pierwsza wersja *MySQL* została stworzona przez szwedzką firmę *MySQL AB*, która 16 stycznia 2008 roku została wykupiona przez *Sun Microsystems*, natomiast *Sun Microsystems* został wykupiony 27 stycznia 2019 roku przez *Oracle.*

System bazodanowy *MySQL* jest znany przede wszystkim ze względu na swoją wydajność i szybkość działania. Języki użyte do zaimplementowania systemu to *C* oraz *C++*, które charakteryzują się bezpośrednim dostępem do funkcji systemowych i zasobów sprzętowych oraz wysoką wydajnością kodu wynikowego. Dodatkowym atutem tego systemu jest jego popularność (drugie miejsce wśród rankingu popularności datującej na marzec 2019 [3]), dzięki czemu większość dostawców usług internetowych uwzględnia *MySQL* w oferowanych pakietach.

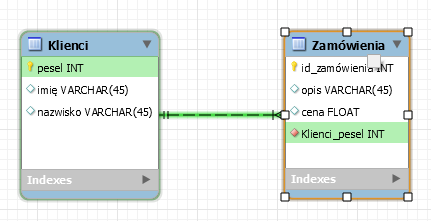
Do głównych cech *MySQL* zalicza się również pełna wielowątkowość, która zapewnia pracę na maszynie wieloprocesorowej, jeśli na takiej system zostanie osadzony. Dołączony mechanizm obsługi klauzul grupujących i agregujących *SQL* czy zastosowanie funkcjonalności indeksów, pozwala zoptymalizować szybkość otrzymywania wyników od sytemu, co jest przydatne w przypadku obszernych baz danych. *MySQL* posiada obsługę transakcji, czyli zbioru operacji na bazie danych, które stanowią całość i zostaną wykonane wszystkie albo żadna z nich. Dzięki temu zostaje spełniona zasada *ACID* (ang. *Atomicity, Consistency, Isolation, Durability*) czyli niepodzielność, spójność, izolacja i trwałość danych czy operacji. Warto również wspomnieć, że *MySQL* posiada dużą liczbę typów danych, czyli w rekordach bazy danych można przechowywać zarówno liczby, ciągi znakowe, obiekty binarne, datę i czas czy typy wyliczeniowe.

Baza danych MySQL umożliwia manipulowanie danymi przy wykorzystaniu języka SQL (ang. *Structured Query Language*). System jest oparty na relacjach, gdzie dane są przechowywane w tabelach, a w tabelach znajdują się wiersze (rekordy). Każda tabela zawiera w sobie klucz główny (podstawowy), który jest unikalnym identyfikatorem dla poszczególnego rekordu. Każdy rekord może mieć tylko jeden klucz podstawowy, który nie może być pusty. Na przykład w bazie danych klientów, numer pesel klienta może być unikalnym identyfikatorem użytym do identyfikacji jego danych (Rys. 3).

Użycie *MySQL* wymaga zdefiniowania określonego schematu bazy danych, który zawiera pogrupowane wszystkie obiekty bazy danych. Ze względu na to, że MySQL jest bazą relacyjną, bez zdefiniowania modelu danych nie można rozpocząć manipulacji danymi.

Schemat relacyjnej bazy danych najczęściej przedstawiony jest za pomocą *Diagramu encji* (ang. *Diagram ERD – Entity-Relationship Diagram*). Diagram ERD jest to graficzne przedstawienie relacji pomiędzy jednostkami, które demonstruje konceptualne modele danych.

Prosty diagram ERD przedstawiony poniżej (Rys. 3), zawiera w sobie informacji o relacji pomiędzy jednostką *Klienci*, a *Zamówienia*.



Rys. 3 Przykładowy diagram ERD

Każdy klient zawiera takie dane jak *imię* oraz *nazwisko* oraz identyfikowany jest poprzez unikalny *pesel* (klucz główny). Każde zamówienie zawiera takie informacje jak *opis*, *cena*, *pesel* klienta, który złożył zamówienia (klucz obcy) oraz unikalny klucz główny *id\_zamówienia*. Relacja przedstawiona pomiędzy dwoma encjami informuje, że *Klient* może mieć wiele zamówień, lecz pojedyncze *Zamówienie* może być przypisane tylko do jednego klienta.

Ze względu na swoje funkcjonalności, opisywana baza danych nadaje się do obsługi każdej aplikacji komputerowej czy internetowej, niezależnie od jej wielkości oraz może zostać uruchomiona na szerokiej gamie platform systemowych, takich jak *Linux*, *Windows*, *OS X*, *Solaris*, *FreeBSD* [5]. *MySQL* wchodzi w skład takich stosów technologicznych jak *LAMP* (Linux, Apache, MySQL, Perl/PHP/Python), a mówiąc szerzej *XAMP*, gdzie pierwsza litera X odnosi się do różnych platform (ang. *cross-platform*).

MySQL posiada szereg języków programowania, poprzez które możemy nawiązać połączenie z bazą danych oraz wykonywać operacje manipulowania danymi. Przykładem popularnych języków wspieranych przez *MySQL* są: C / C++, Java, JavaScript (NodeJS), PHP, Python, Haskell, Objective-C i inne [5].

* 1. PostgreSQL

*PostgreSQL* często nazywany również jako   
*Postgres*, jest to system zarządzania bazą danych, który posiada obiektowo – relacyjny model danych oraz korzysta ze strukturalnego języka zapytań *SQL*. *PostgreSQL*, podobnie jak *MySQL* udostępniony jest na zasadzie *wolnego oprogramowania*.

Rys. . Logo PostgreSQL -   
[www.postgresql.org](http://www.postgresql.org). [03 marzec 2019]

Początki systemu sięgają roku 1977, gdzie na Uniwersytecie Kalifornijskim w Berkeley, pod kierownictwem dr Michaela Stonebrakera, rozpoczęto pracę nad projektem *Ingres* (eksportowa baza danych uczelni). W 1985 roku projekt ewaluował do prac badawczych nad obiektowo-relacyjną bazą danych *Postgres* (*post-ingres*), sponsorowanych przez między innymi DARPA (ang. Defense Advanced Research Projects Agency), ARO (ang. *Army Research Office*) czy NSF (ang. *National Science Foundation*). Przez kolejne lata system był ulepszany poprzez dodawanie nowych regół, typów, elementów obiektowych i procedur. Wynikiem projektu było jego wykupienie przez firmę *Informix* i komercjalizacja pod nazwą *Illustra*.

System otrzymywał kolejne wersje produkcyjne, aż do roku 1994, w którym dwaj członkowie zespołu Stonebrakera – *Andrew Yu* i *Jolly Chen* – dodali do systemu interpreter języka SQL. Projekt udostępniono w maju 1995 pod nazwą *Postgre95*. Od 1996 roku dalsze prace nad systemem podjęła społeczność *wolnego oprogramowania* (ang. *Open Source*), zmieniając nieaktualny znacznik roku wydania w nazwie systemu – 95 na SQL, tworząc tym samym nową nazwę *PostgreSQL* oraz organizację *PostgreSQL Global Development Group,* której zadaniem do dnia dzisiejszego (03 marzec 2019) jest koordynacja rozwoju projektu. W 2005 roku system bazodanowy *PostgreSQL* otrzymał oficjalne wsparcie od firmy *Sun Microsystems* (która w 2010 roku została wykupiona przez Oracle).

Od wersji 9.2 wydanej w 2012 roku, *PostgreSQL* ma spore możliwości. Wzrost wydajności, skalowalności oraz elastyczności znacząco wpłynął na popularność systemu (czwarte miejsce wśród rankingu popularności datującej na marzec 2019 [3]). Sam system został zaimplementowany w języku *C*, co wiąże się z wysoką wydajnością kodu wynikowego oraz wydajnym dostępem do zasobów sprzętowych.

Do głównych cech *Postgres’u* można zaliczyć zaawansowaną implementację standardu *SQL,* uwzględniającą między innymi mechanizm auto-inkrementacji czy obsługę różnych typów plików w tym *BLOB* (ang. *Binary Large Objects*) – czyli obszernych plików binarnych takich jak np. pliki graficzne. Ważną cechą systemu jest możliwość implementacji funkcji w języku *C*, które można skompilować do bibliotek dynamicznych. *PostgresSQL* jest łatwy w rozszerzeniu ze względu na posiadanie takich mechanizmów jak *dziedziczenie, typy danych* czy *funkcji*. Powyższe cechy zaliczają system do kategorii obiektowo – relacyjnych, pomimo tego, *Postgres* jest silnie związany z relacjami.

Podobnie jak w przypadku *MySQL*, *PostgreSQL* pozwala na manipulowanie danymi za pomocą języka *SQL*, jest oparty na relacjach, a dane przechowywane są w tabelach, które zawierają rekordy z danymi. Również w tym przypadku do korzystania z bazy danych potrzeba wcześniej zdefiniowanego schematu (patrz pkt 2.1).

*Postgres* jest obsługiwany przez wszystkie popularne systemy operacyjne takie jak *Windows*, *Linux*, *OS X*, *Solaris*, *OpenBSD*, *FreeBSD* i inne [6]. Dostęp do funkcjonalności systemu można uzyskać poprzez takie języki programowania jak C, C++, Java, JavaScript (NodeJS), Python, PHP i inne [6], co daje spore możliwości dla różnych programistów.

* 1. MongoDB

Tab. . Porównanie kluczowych cech wybranych baz danych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nazwa  systemu | Model | Języki  implementacji | Systemy  operacyjne  serwera | Obsługiwane języki  programowania | Wsparcie transakcji |
| MySQL | Relacyjny | C i C++ | FreeBSD Linux  OS X  Solaris  Windows | Ada  C, C#, C++  Delphi, Eiffel  Erlang, Haskell  Java  JavaScript (Node.js)  Objective-C  Perl, PHP  Python, Ruby | Tak |
| PostgreSQL | Obiektowo – relacyjny | C | FreeBSD  HP-UX  Linux  NetBSD  OpenBSD  OS X  Solaris  Unix  Windows | .Net  C, C++  Delphi, Java  JavaScript (Node.js)  Perl, PHP  Python Tcl | Tak |
| MongoDB | Magazyn dokumentów | C++ | Linux  OS X  Solaris  Windows | Actionscript\*  C, C#, C++  Clojure\*  Dart\*, Delphi\*  Erlang  Go\*  Haskell  Java, JavaScript  MatLab\*  Perl, PHP, PowerShell\* Prolog\*, Python, R\* Ruby, Scala, Smalltalk\* | Transakcje ACID z wieloma dokumentami |

\* nieoficjalny sterownik

1. Użyte technologie

Poniższy rozdział zawiera opis użytych w projekcie technologii. Do implementacji logiki aplikacji testowej zostało wykorzystane środowisko uruchomieniowe JavaScript’u – *NodeJS*, wraz z Framework’iem *ExpressJS*. Do wizualizacji wyników jak i stworzeniu interfejsu graficznego do testów został użyty Framework *Angular* w wersji 7 wraz z technologiami:   
*TypeScript, HTML* oraz *CSS*.

* 1. NodeJS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. ExpressJS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. Angular 7

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. TypeScript

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

* 1. GIT

Tab. . Przykład podpisu tabeli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Listing. 1. Początkowe żądanie HTTP

GET /script/Articles/Latest.aspx HTTP/1.1  
Host: www.codeproject.com  
Connection: keep –alive  
Cache -Control: max-age=0  
Accept: text/html ,application/xhtml+xml,application/xml|  
User -Agent: Mozilla/5.0 ...  
Accept -Encoding: gzip ,deflate ,sdch  
Accept -Language: en-US...  
Accept -Charset: windows -1251,utf -8...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus.

Przykład listy numerowanej:

1. Implementacja

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris id dapibus enim. Etiam lobortis pulvinar enim in maximus. Aliquam erat volutpat. Integer maximus est turpis, ut bibendum ligula accumsan et. Ut eget vestibulum libero. Aliquam erat volutpat. Nullam placerat mauris a lectus tincidunt, et aliquet turpis aliquam. Etiam in malesuada lacus. Proin dignissim augue sit amet auctor elementum. Suspendisse potenti. Vivamus suscipit vulputate massa ac molestie. Suspendisse a justo porttitor, commodo mi at, placerat risus. Integer lobortis augue ac neque suscipit, vel sodales lacus fringilla.

3 i 4 polaczyc: opis środowiska testowego/platformy badawczej, rozwiazania jakie uzylem, technologie, architektura

* 1. Założenia projektowe
  2. Wykorzystane technologie i narzędzia
  3. Opis działania systemu

1. Analiza wydajnościowa
2. Podsumowania

# Literatura

[1] M. Bickley, C. Slominski. A MySQL-based data archiver: preliminary results. Proceedings of ICALEPCS07, Paz. 2007. http://www.osti.gov/scitech/servlets/purl/922267 [dostęp dnia 20 czerwca 2015].

[2] J. Jędrzejczyk, B. Sródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

[1] „Dictionary by Merriam-Webster: America’s most-trusted online dictionary”. [Online]. Dostępne na: https://www.merriam-webster.com/. [Udostępniono: 29-mar-2019].

[2] C. Coronel i S. Morris, *Database Systems*. 2004.

[3] „DB-Engines - Knowledge Base of Relational and NoSQL Database Management Systems”. [Online]. Dostępne na: https://db-engines.com/en/. [Udostępniono: 25-mar-2019].

[4] „Free software is a matter of liberty, not price — Free Software Foundation — working together for free software”. [Online]. Dostępne na: https://www.fsf.org/about/. [Udostępniono: 28-mar-2019].

[5] „MySQL System Properties”. [Online]. Dostępne na: https://db-engines.com/en/system/MySQL. [Udostępniono: 03-kwi-2019].

[6] „PostgreSQL System Properties”. [Online]. Dostępne na: https://db-engines.com/en/system/PostgreSQL. [Udostępniono: 05-kwi-2019].

# Dodatek A