

•



UNIWERSYTET MARII CURIE-SKŁODOWSKIEJ
W LUBLINIE
Wydział Matematyki, Fizyki i Informatyki

Kierunek: Informatyka

Adam Wadowski

nr albumu: 303841

**Porównanie wydajności relacyjnych i nierelacyjnych
baz danych: Firebase i Java Derby, na przykładzie
aplikacji wyszukiwającej gry według zadanych
preferencji w technologii Spring Boot**

Performance comparison of relative and non relative databases: Firebase and
Java Derby, on example of application searching games according to given
preferences in Spring Boot technology

Praca licencjacka

napisana w Zakładzie JAKIMŚ

pod kierunkiem dr. Anny Gajos-Balińskiej

Lublin rok 2023

Spis treści

Wstęp	1
1. Tworzenie aplikacji webowej	3
1.1. Problem wyboru technologii	3
2. Integracja z aplikacją webową	5
2.1. Sekcja	5
3. Zbieranie danych badawczych	7
3.1. Sekcja	7
Podsumowanie	9

Wstęp

Wielu programistów na początku budowy projektu, zderza się z problemem wyboru najbardziej wydajnej bazy danych. Programiści oczekują często jak najmniejszego czasu oczekiwania nawołane przez nich zapytania, łatwej obsługi oraz dobrej integracji z ich aplikacjami, które tworzą do użytku masowego przez wielu użytkowników, przez co bazy danych zwiększają swoją objętość, a ich przeszukiwanie staje się coraz dłuższe. Osobiście borykałem się z tym wyborem już wiele razy, podczas budowy aplikacji w trakcie studiów, życia prywatnego, a także zawodowego. Najczęstszym problemem pojawiającym się na samym początku, jest rozpoznanie, która baza danych będzie najbardziej optymalna dla naszej aplikacji, czy będzie to baza relacyjna, czy może nierelacyjna, jednakże na początku drogi programisty nikt nie jest w stanie dokonać najlepszego wyboru bez odpowiedniego rozeznania i pomocy od osób trzecich.

Celem przedstawionej pracy, jest pokazanie różnic w tworzeniu, implementacji oraz wydajności bazy relacyjnej i nierelacyjnej, na podstawie aplikacji webowej pisanej z użyciem technologii Spring Boot.

Rozdział 1

Tworzenie aplikacji webowej

1.1. Problem wyboru technologii

Wielu programistów podczas tworzenia nowej aplikacji, zastanawia się na początku jaki język programowania wybrać, aby rozwiązać problem jak najlepiej. Obecnie do tworzenia aplikacji webowych używamy wiele technologii takich jak najpopularniejszy JavaScript, HTML/CSS, Python, SQL oraz Java. Aby przełożyć kod na coś widocznego, potrzebujemy kompilatora, który różni się w zależności od używanego języka. Dla przykładu używając języka Java naszym kompilatorem będzie javac. Korzystając z języka jakim jest Java, możemy używać wielu frameworków. Do stworzenia aplikacji internetowej najbardziej popularnym wyborem wśród programistów jest Spring Boot. Głównym powodem wyboru wyżej wymienionego narzędzia jest szybkość budowania aplikacji, w kilka minut możemy wygenerować działającą aplikację i opublikować ją na prywatnym serwerze. Wraz z szybkością idą: obszerna i ogólnodostępna dokumentacja, która potrafi poprowadzić krok po kroku nawet największego laika, możliwość prostego testowania oraz modularyzacja naszego projektu, dzięki której możemy w łatwy sposób zmieniać zależności dla konkretnego kontekstu.

Przy tworzeniu aplikacji webowej, język Java, służy nam jedynie do wystawienia tzw. kontrolerów, które pod określonym adresem kryją swoje funkcjonalności. Do odczytu oraz użycia takiego kontrolera potrzebny nam drugi język programowania, który stworzy widok dla danych pobranych z konkretnej ścieżki. Najpopularniejszym językiem stosowanym w tym momencie do współpracy z Javą jest Angular, który jest prężnie rozwijany przez firmę Google. Jest to framework napisany w języku TypeScript, którego zaletami są między innymi szybkość użycia, wydajność oraz właśnie wymieniona wyżej budowa z TypeScriptu. Pisanie w tym frameworku, polega głównie na tworzeniu komponentów, które są modyfikowane zależnie od danych oraz logiki programu. W ten sposób jesteśmy w stanie stworzyć stronę, która nie jest w stanie zbierać danych w żaden możliwy sposób, do gromadzenia danych potrzebujemy bazy.

Bazy danych dzielą się na wiele kategorii. Z uwagi na miejsce inicjalizacji możemy rozmawiać o lokalnych bazach danych lub typu klient-serwer. W pierwszym rodzaju są to najprostsze zbiory, które są gromadzone na jednym komputerze, a wszelkie zmiany będzie nanosił użytkownik. Drugi rodzaj, który jest przechowywany w zasobach serwera, jest traktowany jako osobny komputer, dostęp do niego możemy uzyskać poprzez połączenie sieciowe. Ze względu na architekturę, wyróżniamy dwa typy: jednowarstwowe oraz dwuwarstwowe. Bazy jednowarstwowe wykonują zmiany od razu, w przeciwieństwie do drugiego typu, w którym połączenie z serwerem odbywa się za pomocą specjalnego sterownika, a kontrolowanie danych zależy od klienta. Ostatnim podziałem baz danych jest podział względem struktur danych, które używają. Jedne zwane prostymi lub kartotekowymi, określają każdą tablicę jako osobny dokument, przez co nie ma między nimi żadnego połączenia. Jedyną wspólną cechą tabel jest stosowanie ich w jednym wspólnym celu. Drugie zwane relacyjnymi bazami danych określają wiele tablic, które mogą się łączyć, ten typ posiada język programowania zwany SQL, który pozwala na tworzenie zaawansowanych funkcji obsługi danych. Wartości używane w tabelach powinny być oparte na prostych typach danych, tabele są dwuwymiarowe, mające zero lub więcej wierszy oraz minimum jedną kolumnę. Możliwe jest również porównywanie wartości w bazie, oraz ich wiązanie w celu wyświetlenia wielu danych

dla jednego rekordu np. dla książki możemy wyświetlić jej zawartość ale również z innej tabeli możemy wyświetlić bibliotekę, w której się znajduje. Do takiej operacji potrzebujemy unikalnego kodu oraz relacji między tabelami. Trzecim typem są bazy obiektowe, które są zdefiniowane tylko jednym standardem z 1993 roku. Podstawowym celem tego modelu jest bezpośrednie odwzorowanie obiektów oraz powiązań między nimi występujących w aplikacji. Ostatnim typem są strumieniowe bazy danych, przedstawiają one dane w postaci zbioru strumieni. System zarządzania jest nazwany strumieniowym systemem zarządzania danymi(ang. Data Stream Management System). Jest to nowy typ, który znajduje się w fazie prototypowej, nie istnieją dla niego żadne rozwiązania komercyjne.

1.2. Projektowanie aplikacji

Każdą aplikację należy zaprojektować, ułatwia to nie tylko pisanie kodu w dalszej części, ale również organizację oraz wskazuje rozwiązania, które programista powinien używać w pracy nad oprogramowaniem. Planowanie budowy powinniśmy podzielić na sześć konkretnych etapów, aby doprowadzić projekt do finalnej wersji oraz wspierać jego rozwój i utrzymanie.

Pierwszą fazą jest czas planowania i analizy. W tej fazie analitycy spotykają się z klientem i rozmawiają na temat oczekiwanych funkcjonalności, wymagań oraz oczekiwań względem wydajności. Jeśli tworzymy aplikację samodzielnie w celach własnych, również należy zastanowić się chwilę nad tymi trzema punktami, a także zapisać je w celu użycia w kolejnych fazach budowy projektu. Powstaje w ten sposób ogólny zarys projektu, Druga faza skupia się na projektowaniu UX/UI, co oznacza przygotowanie projektu oprogramowania z uwzględnieniem specyfikacji z fazy pierwszej. Pomaga to przede wszystkim w ustaleniu funkcjonalności oraz określeniu wyglądu aplikacji, czyli interfejsu użytkownika. Najważniejsze w tym etapie jest to, aby stworzyć interfejs intuicyjny, interaktywny oraz przyjazny dla użytkownika. Faza trzecia czyli tworzenie, jest to najdłuższa faza całego procesu, polega na utworzeniu działającego oprogramowania według zaleceń z poprzednich faz. Faza czwarta czyli testowanie polega na sprawdzeniu, czy programiści uzyskali oczekiwany przez klienta efekt, sprawdzenie wszystkich usług, na które pozwala aplikacja oraz naprawienie potencjalnych błędów. Przed ostatnią fazą polega na wdrożeniu do klienta, nasze oprogramowanie powinno spełniać wszystkie wymagania z fazy pierwszej, a także posiadać interfejs zalecany podczas projektowania. W ostatniej fazie programiści oraz testerzy skupiają się na utrzymaniu projektu, gdy klient zwraca się po nową funkcjonalność, programiści wracają do fazy trzeciej, powtarzając również fazę czwartą i piątą. W tym momencie ważne jest również monitorowanie funkcjonowania aplikacji i analiza, czy wszystko przebiega pomyślnie.

Wytwarzanie oprogramowania, ciągle jest rozwijane, ponieważ wielu firmom zależy na jak najszybszym dostarczeniu gotowej aplikacji do klienta, co nie zawsze jest dobrym rozwiązaniem. Priorytetową sprawą jest przeanalizowanie wymagań postawionych przez klienta. Rozwijane technologie tworzenia oprogramowania, polegają na opracowywaniu nowych bibliotek do języków programowania, nowych frameworków lub całych języków, które dążą do automatyzacji działań programistów.

Rozdział 2

Integracja z aplikacją webową

Tutaj jakiś wstęp do rozdziału.

2.1. Sekcja

Tu sekcja.

2.1.1. Podsekcja

Tu podsekcja. Z tabelką.

1	3.196622222	2.581588889	4.150291803	8.154227869
2	3.34634188	2.45621453	4.479345902	7.572304918
3	2.862333333	2.087170085	3.834808197	6.379259016

Tabela 2.1: Podpis

Rozdział 3

Zbieranie danych badawczych

Tekst.

3.1. Sekcja

Zacytowane [1].

3.1.1. Podsekcja

Tekst.

Podsumowanie

Tekst.

Bibliografia

[1] AUTOR, *TYTUŁ*, WYDAWNICTWO, ROK.

Spis tabel

2.1. Podpis	5
-----------------------	---

Spis rysunków