Projekt nieRelacyjnej Bazy danych

System zarządzania miejscami w akademiku

Filip Maciejak (260375)

Piotr Grygoruk (260299)

Jakub Szwedowicz (243416)

Spis treści

[1. Podsumowanie projektu 3](#_Toc120136818)

[1.1. Analiza projektu 3](#_Toc120136819)

[1.1.1. Początkowe założenia 3](#_Toc120136820)

[1.1.2. Początkowe ograniczenia 3](#_Toc120136821)

[1.2. Kierunki rozwoju 3](#_Toc120136822)

# Podsumowanie projektu

## Analiza porównawcza

### Główne różnice w projektach zrealizowanych za pomocą bazy relacyjnej i nierelacyjnej?

Struktura nierelacyjnej bazy danych jest generalnie prostsza. Wiele tabel zostało połączonych we wspólne dokumenty. Preferuje się redundancję danych ponad większą liczbę zapytań.

### Czy ich zastosowania są tożsame i można je traktować wymiennie?

Zastosowania relacyjnych i nierelacyjnych baz danych są zupełnie różne. Relacyjne bazy danych zakładają rzadko lub wcale zmieniające się struktury tabel, wysoką oszczędność zajmowanego miejsca kosztem większą liczbą zapytań i scentralizowany charakter bazy danych. Nierelacyjne bazy danych mają elastyczną strukturę pozwalającą z łatwością dodawać nowe klucze i wartości do już istniejących dokumentów. Oszczędność przestrzeni dyskowej schodzi na dalszy plan poprzez powszechną redundancję danych. Pozwala to na znacznie prostsza konstrukcję zapytań. Bazy takie również są przystosowane do systemów IoT z racji na implementację pozwalającą na shardowanie serwerów i wzrost horyzontalny.

### W jakich zastosowaniach lepiej sprawdzają się bazy relacyjne, a w jakich wybrana technologia nierelacyjna?

Relacyjne bazy danych dobrze sprawdzają się modelując przewidywalne i niezmienne środowiska. Architekturę ogranicza również możliwy wyłącznie wzrost wertykalny. Bazy nierelacyjne są pod tym względem znacznie elastyczniejsze. Zostały zaprojektowane z myślą o równolegle działających serwerach zatem oferują możliwość zwiększenia wydajności bazy danych zarówno poprzez dokupienia kolejnych serwerów jak i poprzez usprawnienie parametrów istniejących. Ponad to z racji na oparcie swojego działania na dokumentach (przeważnie json) pozwalają również na ciągłe poszerzenia zawartości dokumentów o kolejne klucze. Jest to przydatna właściwość w dynamicznie rozwijających się projektach, w których ciężko jest przewidzieć przyszłe zapotrzebowanie oraz architekturę.

### Czy pojawiła się konieczność zmiany założeń lub wybrane wymagania były niemożliwe do zrealizowania

Relacyjną bazę danych udało się w całości przenieść do nierelacyjnej.

### W jaki sposób wybrana baza NoSQL różni się od bazy relacyjnej pod względem definiowania zapytań

Zapytania są generalnie prostsze w napisaniu i funkcjonalnie nie różnią się szczególnie od popularnych języków programowania. Pozwala to również łatwo połączyć funkcjonalności bazy danych z istniejącym kodem produkcyjnym.

### Jakie są główne różnice w wydajności i jakie są ich przyczyny

Bazy nierelacyjne zajmują więcej miejsca z racji na wykorzystanie mało oszczędnego formatu .json. Ponad to same systemy (MongoDB, Cassandra, itd.) zostały opracowane z myślą o wielokrotnym powielaniu danych na wypadek awarii serwera. Zapytania są również często szybsze ponieważ najczęściej dotyczą już połączonych we wspólny dokument danych i w związku z tym nie ma konieczności przeszukiwania kilku różnych tabel dla jednego zapytania. Sam charakter pracy z bazami nierelacyjnymi (wykorzystanie .jsonów i promowanie szybkości ponad zajmowane miejsce) sprawia, że dane są również często powielane w obrębie kilku tabel (tzw. redundancja danych), gdyż uznaje się że w środowiskach w jakich są one wykorzystywane, lepiej jest duplikować dane w różnych dokumentach i otrzymywać zawsze w json-ach cały komplet potrzebnych informacji, niż zbierać ulokowane w różnych częściach systemu dane dla każdego zapytania.