

ПОРІВНЯЛЬНИЙ АНАЛІЗ РЕЛЯЦІЙНИХ ТА НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ НА ОСНОВІ ВЕБДОДАТКУ, РОЗРОБЛЕНОГО МОВОЮ PYTHON

Никитюк Ангеліна

COMPARATIVE ANALYSIS OF RELATIONAL AND NON-RELATIONAL
DATABASES BASED ON A WEB APPLICATION DEVELOPED IN PYTHON

Angelina Nykytiuk

Волинський національний університет імені Лесі Українки, просп. Волі, 13, Луцьк,
43025, Україна

Abstract. *The study explores approaches to data storage in modern web applications using both relational and non-relational databases. A hybrid architecture was implemented, combining PostgreSQL for structured information and MongoDB for handling nested comments. Performance testing of both DBMSs was conducted, confirming their efficiency in their respective use cases. The results demonstrated the viability of a combined approach for building flexible, scalable, and reliable information systems.*

У процесі створення сучасних інформаційних систем одним із ключових завдань є вибір оптимальної моделі зберігання та обробки даних. Найпоширенішими підходами залишаються реляційні та нереляційні бази даних, кожна з яких має свої переваги, обмеження та специфіку застосування. Реляційні бази даних забезпечують організацію даних у вигляді таблиць із строго визначеною схемою, що дозволяє зберігати структуровану інформацію з гарантією цілісності. Їх ключовою особливістю є підтримка зв'язків між таблицями через зовнішні ключі, що дає змогу створювати складні логічні структури даних. У свою чергу, нереляційні бази – це сучасна альтернатива, орієнтована на роботу з динамічними, напівструктурованими або неструктурованими даними, де традиційна модель таблиць не є ефективною [2].

У роботі розроблений вебдодаток, який дозволяє користувачам реєструватися, створювати та редагувати публікації, залишати коментарі, а також переглядати контент інших авторів. У цьому проєкті було реалізовано гібридну архітектуру, що поєднує реляційний механізм управління даними з гнучкістю NoSQL. PostgreSQL виступає в ролі основної СУБД, де зберігаються облікові записи користувачів, записи блогу, мітки часу та інші структуровані сутності. Цей вибір пояснюється високим рівнем підтримки ACID-властивостей – атомарності, узгодженості, ізоляваності та надійності транзакцій [3]. PostgreSQL також має розширену підтримку JSON, що дає змогу частково працювати з напівструктурованими даними, що підсилює її універсальність [6]. Використання SQL у PostgreSQL забезпечує можливість створення складних аналітичних запитів, фільтрації, агрегування та модифікації даних з високою точністю.

Для обробки коментарів, які мають вкладену або динамічну структуру, у вебдодатку було інтегровано документо-орієнтовану базу даних MongoDB. Ця СУБД підтримує зберігання даних у форматі BSON, що є розширенням JSON, і дозволяє розміщувати вкладені об'єкти без попередньої схеми [5]. MongoDB ідеально підходить для вебдодатків, де структура даних часто змінюється, як-от у коментарях, що можуть містити відповіді, вкладення або змінювану кількість полів. Її модель дозволяє реалізовувати горизонтальне масштабування,

зберігаючи при цьому високу продуктивність [4]. Крім того, MongoDB підтримує вбудовані індекси, що значно прискорює виконання запитів. Це робить її особливо придатною для модулів, де гнучкість і швидкість мають вищий пріоритет, ніж строгість структурованості.

У ході експериментального тестування було виявлено, що PostgreSQL демонструє стабільну роботу при виконанні транзакцій навіть під високим навантаженням. Це пояснюється наявністю повноцінної підтримки транзакцій та забезпеченням консистентності даних. Завдяки строгому контролю над схемами даних реляційна модель мінімізує помилки та забезпечує цілісність у фінансових, освітніх та корпоративних системах. Однак, при реалізації динамічних функціональностей, таких як коментування, її жорстка схема даних вимагає складніших маніпуляцій для внесення змін до структури.

Саме в таких випадках проявляються переваги MongoDB. Вона дозволяє швидко адаптувати структуру колекцій, що дає змогу скоротити час розробки та покращити взаємодію користувача з інтерфейсом додатку. Завдяки відсутності вимог до попередньої структури таблиць та можливості зберігати вкладені об'єкти, MongoDB надає більшу гнучкість у роботі з контентом користувача. Крім MongoDB, до NoSQL-сімейства належать такі рішення, як Redis (для кешування) та Kafka (для обробки потоків даних), що часто застосовуються в реальних системах для оптимізації швидкодії або асинхронної обробки подій [1].

Тестування PostgreSQL охоплювало перевірку транзакційних властивостей системи, коректності збереження та оновлення даних, дотримання зовнішніх ключів, а також впливу індексів на продуктивність обробки запитів. MongoDB, яка використовується для зберігання коментарів, протестовано на здатність обробляти вкладені документи, змінну структуру записів та вплив індексів на швидкість обробки запитів. Було підтверджено, що MongoDB ефективно справляється з навантаженням, демонструючи високу швидкість доступу до даних, що особливо важливо для модулів, які потребують динамічного оновлення інформації у режимі реального часу.

Результати дослідження підтверджують, що ефективна архітектура сучасного вебдодатку може ґрунтуватися на гібридному підході до зберігання даних. Реляційні бази, зокрема PostgreSQL, залишаються незамінними у задачах, де потрібна чітка структура та гарантії цілісності, тоді як нереляційні рішення, як MongoDB, забезпечують масштабованість, гнучкість і швидко реакцію на зміни даних. Такий підхід дає змогу створювати стійкі, продуктивні та зручні для користувача інформаційні системи, здатні адаптуватися до сучасних вимог бізнесу та технологій.

Бібліографія

1. Bansal P. Kafka: powering real-time data streaming, its limitations, and learning resources. URL: <https://medium.com/@prateekbansalind/kafka-powering-real-time-data-str>.
2. Code With C. How NoSQL And SQL Will Work Together (And Why That Matters). URL: <https://www.codewithc.com/how-nosql-and-sql-will-work-together-and-why>.
3. Database.Guide. What does ACID mean in Database Systems?. URL: <https://database.guide/what-is-acid-in-databases/>.
4. Infonikhil. SQL vs NoSQL: Which Database is Best for Horizontal Scaling, Sharding, and Replication?. URL: <https://medium.com/@infonikhil270/sql-vs-nosql-which-database-is-best->.
5. Patra S. K. NoSQL Databases Overview: Types, Use Cases, and Key Differences. URL: <https://www.csinfo360.com/2024/09/nosql-databases-overview-types-use.h>.
6. PostgreSQL Global Development Group. 9.16. JSON Functions and Operators. URL: <https://www.postgresql.org/docs/current/functions-json.html>.