

ОГЛЯД ДОКУМЕНТО-ОРІЄНТОВАНИХ БАЗ ДАНИХ НА ПРИКЛАДІ MONGODB

¹ Вінницький національний технічний університет;

Анотація

Зроблено огляд документо-орієнтованих баз даних та способів роботи з ними на прикладі документо-орієнтованої бази даних MongoDB.

Ключові слова: JSON, веб-додатки, MongoDB, розробка веб-додатків, SQL, СКБД.

Abstract

An overview of document oriented databases and methods of work with it on the example of a document oriented MongoDB database.

Keywords: JSON, web-applications, MongoDB, web-applications development, SQL, DBMS.

Вступ

На сьогоднішній день розробка додатків для будь-яких платформ, в тому числі і для веб, в більшості не обходиться без використання баз даних, так як в багатьох з них потрібно зберігати дані користувача, дані про проекти, дані про якісь об'єкти та ін.. Часто цими базами даних є реляційні бази даних, такі як PostgreSQL, MySQL, Microsoft SQL та ін., які зберігають інформацію у вигляді таблиць. Але з кожним днем набирають своє популярності так звані документо-орієнтовані бази даних, які використовують ієрархічні структури для зберігання даних. В основі цих баз даних лежать документні сховища, що мають структуру дерева, а іноді структуру лісу.

Прикладом запису в документо-орієнтованій базі даних є такий запис:

```
{  
  Ім'я: «Микола»,  
  Місто: «Вінниця»,  
  Професія: «Веб-розробник»  
}
```

Результати дослідження

Центральним поняттям документо-орієнтованих баз даних є *Документ*. Деталі визначення цієї головної концепції різняться у різних реалізаціях документо-орієнтованих баз даних, але загалом всі вони мають на увазі, що документ ізолює дані та кодує їх (або інформацію) у певному форматі або кодуванні. Це кодування може використовувати XML, YAML, JSON, BSON, а також бінарні формати, такі як PDF чи документи Microsoft Office (MS Word, Excel, тощо).

В MongoDB кодування найчастіше використовує формат JSON (JavaScript Object Notation), так як цей формат є одним з найчастіше використовуваних в роботі з веб-додатками і не тільки, а також він є доволі зрозумілим адже цей формат базується на тексті та може бути прочитаний людиною. Формат JSON може описувати об'єкти та інші структури даних. Переважно використовується для передачі структурованої інформації в мережі. JSON будується на таких двох структурах:

- Набір пар: ключ/значення. Багато в яких мовах це реалізовано як звичайний об'єкт.
- Впорядкований список значень. В багатьох мовах реалізовано як масив, список або послідовність

Для роботи з JSON форматом у мові JavaScript є такі методи, як *eval()*, але так, як він є доволі небезпечним, був запропонований новий метод *parseJSON()* або *JSON.parse()*, але вони здатні обробляти лише JSON дані.

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

Основні можливості MongoDB:

- Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних)
- Досить гнучка мова для формування запитів
- Динамічні запити
- Повна підтримка індексів
- Профілювання запитів
- Швидкі оновлення «на місці»
- Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео
- Журналювання операцій, що модифікують дані в БД
- Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг
- Може працювати відповідно до парадигми MapReduce

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS. Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Для початку роботи з MongoDB в парі з Node.js потрібно надзвичайно мало часу, адже для підключення до бази даних достатньо лише декілька стрічок коду:

```
var mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/db_name');
```

Також всі функції CRUD (Create, Read, Update, Delete – укр. Створити, Зчитати, Оновити, Видалити) на Node.js з MongoDB виглядають надзвичайно просто та лаконічно:

- Створити:
// один користувач
\$ db.users.save({ name: 'Chris' });
// декілька користувачів
\$ db.users.save({ name: 'Chris' }, { name: 'Holly' });
- Зчитати:
// показати всіх користувачів
\$ db.users.find();
// знайти певного користувача
\$ db.users.find({ name: 'Holly' });
- Оновити:
db.users.update({ name: 'Holly' }, { name: 'Holly Lloyd' });
- Видалити:
// видалити все

```
db.users.remove({ });  
// видалити певного користувача  
db.users.remove({ name: 'Holly' });
```

Висновки

Встановлено, що використання баз даних є майже невід'ємною частиною розробки додатків, як для комп'ютера так і для веб. Є два найбільш розповсюджених типи баз даних – реляційні та документо-орієнтовані. MongoDB в поєднанні з Node.js в результаті дає надзвичайно простий та зрозумілий інтерфейс для управління даним, які легко читати користувачу, який не має багато досвіду в користуванні базами даних. Саме завдяки цьому, документо-орієнтовані бази даних, а особливо MongoDB, набирають великої популярності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Документо-орієнтована система керування базами даних [Електронний ресурс]. [Веб-сайт]. – 2022. – Режим доступу до ресурсу: https://www.wiki.uk-ua.nina.az/Document-oriented_database.html.
2. JSON [Електронний ресурс]: [Веб-сайт] .– 2022. – Режим доступу: <https://www.wiki.uk-ua.nina.az/JSON.html>.
3. MongoDB [Електронний ресурс]: [Веб-сайт] .– 2022. – Режим доступу: <https://www.wiki.uk-ua.nina.az/MongoDB.html>.
4. Знайомство з MongoDB [Електронний ресурс]: [Веб-сайт] .– 2022. – Режим доступу: <https://codeguida.com/post/233>

Кривенко Іван Іванович – студент групи 2КН-18Б, кафедра комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ivan.kryv14@gmail.com

Науковий керівник: **Богач Ілона Віталіївна** – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com

Kryvenko Ivan Ivanovich – Student of 2CS-18b group, Department of Computer Science, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ivan.kryv14@gmail.com

Supervisor: **Bogach Ilona Vitaliivna** – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com