

5. <https://www.section.io/engineering-education/what-is-mvc-and-how-does-it-work/>

6. <https://medium.com/@grover.vinayak0611/what-is-flux-architecture-why-facebook-used-it-and-the-comparison-with-mvc-architecture-49c01ed5d2e1>

УДК 004.652

В.Ю. Каштан¹, О.О. Кваша¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ГІБРИДНИЙ СТАНДАРТ ОРГАНІЗАЦІЇ ДАНИХ НА ОСНОВІ ФОРМАТУ JSON

Анотація. У роботі запропоновано новий стандарт організації даних на основі реляційної бази даних та "NoSQL" баз даних. Це дозволило використати вкладені об'єкти та масиви даних для реалізації зв'язків між записами бази даних в єдиній сутності.

Ключові слова: бази даних, SQL, JSON, NoSQL, MongoDB, MySQL, Maven, Jackson.

Вступ. Сьогодні актуальним є питання зберігання та використання даних. Для економічної та соціальної сфери життєдіяльності суспільства важливим є використання інформаційних технологій з великими обсягами даних. У зв'язку з цим постає завдання оптимальної систематизації та зберігання даних у базах даних [1]. Так, з кінця 80-х років 20-го століття реляційні бази даних були і є провідними на ринку [2, 3], з іншого боку перспективним є використання NoSQL баз даних (БД). Бази даних NoSQL спеціально створені для певних моделей даних і мають гнучкі схеми, що дозволяє розробляти сучасні програми.

Тому, дана робота присвячена дослідженню "NoSQL" та реляційних баз даних і можливість комбінувати їх принципи організації даних для забезпечення максимальної гнучкості та структурованості. Реляційні бази даних мають дуже гарні можливості для підтримки великого проекту, де всі дані чітко структуровані, а "NoSQL" – це інструмент для розробки невеликих проектів з великими темпами розвитку, де кількість полів може бути змінна для кожного запису.

Метою даної роботи є розробка стандарту організації даних, що має обов'язкові поля для ідентифікації записів, використання складних об'єктів в якості полів для реалізації зв'язків між записами одної сутності. Для реалізації стандарту був використаний текстовий формат обміну даними JSON.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- визначити обов'язкові поля для кожного елемента;

– визначити можливі обмеження для розробленого стандарту при подальшому використанні;

– визначити можливості використання в сучасних проектах.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти:

- Maven;
- Jackson;
- MySQL;
- MongoDB;
- IntelliJ IDEA;
- Android Studio.

Реляційні бази даних надають гарні можливості зі сторони структурованості даних та продуктивності всієї системи. Але ці бази даних потребують використання зв'язків між окремими сутностями (таблицями), що є базовим механізмом цих баз даних [4]. Тобто, для складних рекурсивних зв'язків між записами одної таблиці потрібно використовувати принаймні дві таблиці. "NoSQL" бази даних більш гнучкі, вони можуть навіть не використовувати первинні ключі для доступу до записів. Для зв'язків між колекціями документів в MongoDB [5] використовують ObjectId [6]. Цей підхід дуже корисний через відсутність JOIN-запитів та масштабування даних [7] (рис.1), але він також вимагає створення додаткової колекції, а через можливу відсутність унікальних цифрових ідентифікаторів документів може зменшуватися продуктивність всієї системи.

Тому, в деяких випадках організація даних вимагає специфічного підходу для зберігання зв'язків між однаковими записами, не створюючи нові сутності.

```
{
  "id" : 0,
  "property1" : "Some data",
  "Categories" : [ {
    "Category1" : 1
  }, {
    "Category2" : -1
  } ]
}, {
  "id" : 1,
  "property1" : "Other data",
  "Categories" : [ {
    "Category1" : -1
  }, {
    "Category2" : -1
  } ]
} ]
```

Рис. 1. Приклад організації даних на основі JSON-файлу

На рисунку 1 представлено фрагмент БД, де дані представлено у вигляді масиву об'єктів. Кожен об'єкт являє собою один запис до бази даних. Він може мати безліч властивостей (колонок). Серед властивостей обов'язкові всього дві: "id" та "Categories". "id" – це простий, не від'ємний, унікальний, цифровий

ідентифікатор. Він дозволяє швидко та зручно отримувати доступ до будь-якого існуючого запису. "Categories" – це масив об'єктів кожен з яких відтворює зв'язки між записами. В цих об'єктів може бути безліч властивостей але повинна бути хоча б одна властивість, яка містить ціле число, що буде мати значення ідентифікатору на інший запис. Для зручності ці поля в даному випадку мають назви: "Category1" та "Category2". Якщо значення ідентифікатору від'ємне, тоді зв'язок по цій категорії з іншим записом відсутній. Категорій може бути будь яка кількість.

Даний підхід може стати зручним засобом обміну інформацією для клієнт-серверних додатків за рахунок де-нормалізації даних, структурованості даних, використання числових ідентифікаторів, зберігання всієї інформації в єдиній сутності. За рахунок базування на форматі файлів JSON такий стандарт може бути легко представлений документами системи керування базами даних MongoDB. Завдяки такій сумісності з MongoDB можливе використання засобів покращення працездатності серверів за рахунок індексування, "шардінгу" або концепції master-slave.

Наукова новизна полягає у створенні нового стандарту організації даних для надання можливості використання рекурсивних зв'язків в одній сутності. Це дозволило забезпечити гнучке використання де-нормалізованих даних.

Висновки. Наявні в сучасному світі системи управління базами даних надають широкі можливості для зберігання, обробки, внесення та видалення інформації. Але при використанні структурованих даних мають суттєвий недолік, а саме для збереження всіх зв'язків в одній сутності потрібні виконати додаткові роботи з нормалізації бази даних, створити нові сутності та відтворити зв'язки між ними. Тому, в дані роботі запропоновано використати "NoSQL" бази даних на основі не структурованих даних. Це дозволило застосувати до кожного запису різну кількість полів, що полегшує роботу з даними при розробці проектів та надає можливість використовувати вкладені об'єкти та масиви даних для реалізації зв'язків між записами в єдиній сутності.

ПЕРЕЛІК ПОСИЛАНЬ

- 1.Oracle documentation [Електронний ресурс] – Режим доступу: <https://www.oracle.com/database/what-is-a-relational-database/>
- 2..OraclePatches [Електронний ресурс]: Режим доступу: <https://qps.ru/8zyo5> - Реляційна модель у порівнянні з документоорієнтованою моделлю. – (Дата звернення 28.05.2019).
- 3.TagLine [Електронний ресурс]: Режим доступу: <https://tagline.ru/database-management-systems-rating/> – Рейтинг систем управління базами даних (СУБД) 2016. – (Дата звернення 28.05.2019).
- 4.Data & Analytics. Database Basics: Concepts & Examples for Beginners [Електронний ресурс] – Режим доступу: <https://www.lido.app/post/database-101>
- 5.Документація MongoDB. Introduction to MongoDB. [Електронний ресурс] – Режим доступу: <https://docs.mongodb.com/manual/introduction/>

6.Types of Database Management Systems [Електронний ресурс] – Режим доступу: <https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/>

7.What is Normalization in DBMS (SQL)? 1NF, 2NF, 3NF, BCNF Database with Example [Електронний ресурс] – Режим доступу: <https://www.guru99.com/database-normalization.html>

УДК 004.8

Н.Р. Книрик¹, Т.С. Заїченко¹

¹Національний університет кораблебудування імені адмірала Макарова,
Миколаїв, Україна

РОЗРОБКА АЛГОРИТМУ ІДЕНТИФІКАЦІЇ ТА ОБРОБКИ ГОЛОСОВИХ КОМАНД ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ГОЛОСОВОГО УПРАВЛІННЯ

Анотація. Створення додатку голосового асистента дає змогу полегшити голосове управління комп'ютерами. На даний момент існує низка проблем, з якими зустрічаються користувачі, а саме обмеженість використання голосового асистента в деяких країнах, можливість роботи такого додатку на комп'ютері, погане розуміння мови та можливість працювати з додатком без підключення до мережі Internet.

Ключові слова: *голосовий асистент, розпізнання мови, Alexa, Google, Siri, Cortana, Python, голосове управління.*

Вступ. В ході дослідження було виявлено, що на сьогоднішній час широко використовуються голосові асистенти, адже вони спроможні економити час користувача на пошук інформації в ситуаціях коли потрібна мультизадачність користувача в роботі. На даний момент тема створення та розвитку системи голосового управління є актуальною, оскільки все більше технологій можуть працювати с голосовим помічником, починаючи від мобільних пристроїв та розумної техніки та закінчуючи машинами на автопілоті.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані та вирішені такі завдання:

- проаналізувати особливості обробки даних за допомогою голосового асистента;
- виконати порівняльний аналіз існуючих рішень системи голосового управління ПК;
- аналіз структури типової технології голосового управління ПК;
- розробити узагальнену структуру та алгоритмічну базу системи голосового управління ПК;