

Карпенко М.І.

Національний університет харчових технологій

Чумаченко С.М.

Апарат РНБО України

Мошенський А.О.

Національний університет харчових технологій

Гуйда О.Г.

Таврійський національний університет імені В.І. Вернадського

ВИБІР ОПТИМАЛЬНОЇ БАЗИ ДАНИХ ДЛЯ СТВОРЕННЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ВИЯВЛЕННЯ НЕБЕЗПЕЧНИХ РЕЧОВИН

Стаття присвячена дослідженню реляційних та нереляційних баз даних з метою визначення оптимального рішення для інтеграції в майбутній програмно-апаратний комплекс, призначений для виявлення небезпечних речовин. Мета дослідження полягає у виборі найкращої бази даних (БД) для створення програмно-апаратного комплексу виявлення небезпечних речовин. Для досягнення цієї мети було використано різні методи аналізу та синтезу: емпіричний, структурно-генетичний аналіз, контент-аналіз, експериментальна реалізація, зокрема, проведено літературний аналіз наукових публікацій для вивчення особливостей реляційних та нереляційних видів БД, а також виокремлення їх основних переваг та недоліків. Обрано вибірку БД для впровадження у проєкт: MySQL, PostgreSQL, MongoDB, Redis та InfluxDB. Виявлено найбільш оптимальні для проєкту БД: MySQL та MongoDB. Виявлені слабкі та сильні сторони кожної з БД. Для демонстрації практичної реалізації, було успішно здійснено підключення плати на базі ESP8266 до MySQL через API інтерфейсу сервісу «000webhost.com», демонструючи можливість інтеграції IoT-застосунків. В результаті було виявлено, що MySQL є найбільш оптимальним вибором для впровадження у даний комплекс. Такий висновок базується на декількох ключових аспектах, включаючи доступність MySQL завдяки ціновій політиці Oracle, його популярність у галузі, значну підтримку з боку спільноти та наявність документації, що полегшує вирішення потенційних проблем. Особливо важливим фактором у виборі MySQL стало його здатність ефективно функціонувати на малопотужних серверах, що є суттєвим для даного проєкту. Хоча MySQL поступається за швидкодією деяким іншим базам даних, його перевага полягає у здатності до оптимального управління розміром файлів баз даних, що забезпечує високий рівень продуктивності. Додатковою перевагою MySQL є його гнучкість та можливість інтеграції з додатковим обладнанням, зокрема з розширеною пам'яттю PSRAM, яка сприяє більш ефективному опитуванню датчиків і збалансованому обміну даними. Це дозволяє забезпечити надійне та ефективне зберігання даних, що є критичним для успішного розвитку проєкту в майбутньому. Таким чином, MySQL, у поєднанні з PSRAM, становить оптимальне рішення для реалізації програмно-апаратного комплексу, орієнтованого на виявлення небезпечних речовин. Отримані результати мають практичне значення для розробки застосунків, програмно-апаратних комплексів та програм, які мають взаємодіяти з БД. Дана робота може бути корисною для розробників та фахівців, які планують використовувати БД у своїх проєктах.

Ключові слова: MySQL, IoT, PSRAM, ESP8266, Arduino.

Постановка проблеми. У процесі розробки програмно-апаратного комплексу дистанційного зондування небезпечних речовин постала проблема вибору найбільш ефективної БД для зберігання даних сенсорів. Бажаним результатом, який вирішить цю проблему, має бути БД, яка б мала можливість отримувати дані з мікроконтролера

через мережу Інтернет, що дало б нові можливості для розвитку IoT. Основними обмеженнями є ціна, вимогливість до ресурсів та надійність.

Аналіз останніх досліджень і публікацій. Для досягнення цієї мети, було проведено детальний огляд періодичних наукових видань, які вийшли за останні 5 років. Використання БД у наш час

є широко розповсюдженим явищем. Про це свідчать робота «Wireless Biometric Fingerprint Attendance System Using Arduino and MySQL Database» Франсуа Секере та Віллі К. Офосу в 9 випуску Міжнародного журналу комп'ютерних наук, техніки та застосувань (IJCSSEA) [1] та розділ книги Чарльза Белла «MySQL and Arduino: United at Last!» [2]. Особливою популярністю платформа з відкритим кодом для програмування мікроконтролерів Arduino користується в Індії, про що свідчать статті Індонезійських журналу Jurnal Jaringan Telekomunikasi «Sistem pencatatan hasil timbangan menggunakan sensor load cell melalui database berbasis arduino uno» [3] та журналу ILCOM Jurnal Ilmiah «Realtime Database Sensor Menggunakan Arduino UNO Untuk Keperluan Sistem Informasi» [4], у якій йдеться про додавання БД до розумного дому. Найбільш значущим є дослідження «Performance evaluation of IoT data management using MongoDB versus MySQL databases in different cloud environments» [5], у якому порівнюються реляційні та нереляційні БД для досягнення максимальної ефективності по параметрам швидкодії та використання пам'яті.

Постановка завдання. Метою статті є проаналізувати реляційні та нереляційні БД. Обрати найбільш ефективну БД згідно критеріїв: вимогливість до ресурсів, ціна, надійність.

Виклад основного матеріалу дослідження. Бази даних (БД) у порівнянні з іншими рішеннями (Google Sheets, сховищами даних або прямим зберіганням на IoT пристроях) мають ряд переваг:

- Структуроване зберігання: Бази даних надають структурований підхід до зберігання даних, де ви можете визначити таблиці, стовпці та відношення між даними. Це дозволяє організувати дані в логічній формі і спрощує доступ до них.

- Ефективний пошук і фільтрація: Бази даних надають механізми для швидкого пошуку та фільтрації даних за різними критеріями. Ви можете використовувати запити для витягування потрібної інформації з бази даних без необхідності проглядати всі дані вручну.

- Система безпеки: Бази даних дозволяють встановлювати рівні доступу та права на різні дані. Ви можете контролювати, хто має право переглядати, змінювати або видаляти дані з бази.

- Масштабованість: Бази даних дозволяють масштабувати обсяги даних і робити ефективні операції з великими наборами даних. Ви можете додавати нові записи, розширювати схему бази даних і обробляти великі обсяги даних без втрати продуктивності.

- Консистентність даних: Бази даних надають механізми для забезпечення цілісності даних, що означає, що дані залишаються узгодженими та правильними навіть при одночасному доступі декількох користувачів або процесів.

Підключення Esp8266 та Esp32 можна легко реалізувати з наступними БД з відкритим кодом: MySQL, PostgreSQL, MongoDB, Redis та InfluxDB. Ці бази даних підтримують протоколи комунікації (наприклад, HTTP або MQTT), що дозволяють взаємодіяти з мікроконтролером.

Усі з перелічених СУБД мають право на реалізацію у проекті. Основна різниця між ними полягає в їх ліцензіях, планах підтримки та особливостях. Кожна з них може задовольнити потребу у збереженні та обробці даних з датчиків і до кожної існують способи підключення мікроконтролерів. Враховуючи такі фактори як попередній досвід розробки автора, популярність БД та доступність навчальних матеріалів та посібників, варто розглянути детальніше та порівняти MySQL та MongoDB.

MySQL – це одна з найпопулярніших реляційних баз даних, СУБД, розроблена для зберігання та управління великим обсягом даних, базується на мові структурованого запиту (SQL) і надає широкі можливості для організації та маніпулювання даними [5].

У MySQL існують різні типи таблиць (по суті движків БД), які можна використовувати в залежності від потреб проекту. Кожен тип таблиці визначає, як саме дані будуть зберігатися і опрацьовуватися в базі даних. Кожен тип має свої особливості і підходить для різних сценаріїв використання. Основні типи таблиць в MySQL включають: MyISAM, InnoDB, MariaDB, MEMORY, CSV, NDB (MySQL Cluster), тощо.

Однією з найбільш популярних є MariaDB завдяки більш високій продуктивності; можливості відновлення стану з будь-якої точки в журналі операцій, включаючи підтримку CREATE/DROP/RENAME/TRUNCATE; відкату результатів виконання поточної операції у разі переривання та підтримці всіх форматів стовпців MyISAM.

Серед переваг MySQL можна виділити наступні:

- Надійність та стабільність: MySQL забезпечує обробку великої кількості транзакцій і забезпечує цілісність даних навіть при великому навантаженні.

- Швидкодія обробки запитів та доступу до даних. СУБД оптимізована для роботи з великими обсягами даних і може працювати ефективно навіть на серверах з обмеженими ресурсами.

Порівняння БД

Характеристика	MySQL	PostgreSQL	MongoDB	Redis	InfluxDB
Тип бази даних	Реляційна	Реляційна	Документ-орієнтована	Ключ-значення	Часовий ряд
Мова запитів	SQL	SQL	JSON-подібна мова запитів	Команди Redis	InfluxQL
Структура даних	Структурована	Структурована	Неструктурована	Неструктурована	Неструктурована
Підтримка транзакцій	Так	Так	Ні	Ні	Так
Підтримка реплікації	Так	Так	Так	Так	Так
Масштабованість	Горизонтальна та вертикальна	Горизонтальна та вертикальна	Горизонтальна	Горизонтальна та вертикальна	Горизонтальна та вертикальна
Швидкодія	Швидка для багатьох сценаріїв	Швидка та оптимізована	Швидка для запису, повільна для читання	Дуже швидка для простих операцій	Швидка для часових рядів
Складність встановлення та налаштування	Середня	Висока	Середня	Низька	Середня
Спільнота та підтримка	Велика спільнота та активна розробка	Велика спільнота та активна розробка	Велика спільнота та активна розробка	Велика спільнота та активна розробка	Середня спільнота та активна розробка
Безпека	Середня	Висока	Вище середнього	Середня	Середня
Відкритість коду	Так	Так	Так	Так	Так
Вимоги до ресурсів	Середні (працює навіть на малопотужних серверах)	Вище середніх	Середні	Вимагає обмежену кількість пам'яті	Потребує великої кількості пам'яті та процесорної потужності для оптимальної роботи з великими обсягами часових рядів даних

– Легкість освоєння: MySQL має простий та зрозумілий синтаксис SQL, що полегшує розробку та обслуговування бази даних. Окрім того, дана СУБД має велику спільноту користувачів, що забезпечує доступ до документації та підтримки.

– Доступні розширення: MySQL підтримує розширення за допомогою різних движків баз даних, які можуть бути використані для специфічних потреб проекту.

Серед недоліків MySQL можна виділити наступні:

– Масштабованість: MySQL може мати обмеження щодо масштабування при обробці дуже великого обсягу даних або при високому навантаженні. Для роботи з дуже великими базами даних може знадобитись додаткове налаштування та оптимізація.

– Відсутність певних функцій, які присутні в деяких інших СУБД. Наприклад, підтримка повнотекстового пошуку, геопросторових запитів або аналітичних функцій може бути обмеженою в стандартній конфігурації MySQL.

– Управління схемою бази даних в MySQL може бути трохи складнішим порівняно з деякими іншими реляційними СУБД. Додавання, зміна

або видалення структури таблиць може вимагати додаткових зусиль та уваги до деталей.

– Відсутність вбудованої підтримки JSON: підтримка роботи з JSON-даними в MySQL була впроваджена пізніше, починаючи з версії 5.7. Якщо вам потрібно широко використовувати JSON-структури в базі даних, можуть бути кращі альтернативи, що мають більш розширену підтримку JSON.

– Безпека: функції безпеки MySQL не такі надійні, як в інших системах, що робить її вразливою до атак, якщо її не захищено належним чином.

Проте, дані проблеми легко вирішуються за допомогою встановлення необхідного ПЗ:

Для роботи з дуже великими базами даних або при високому навантаженні можна використовувати розподілені системи баз даних, такі як MySQL Cluster або MySQL InnoDB Cluster.

Для розширення функціональності MySQL можна використовувати розширення та плагіни. Наприклад, для повнотекстового пошуку можна встановити розширення MySQL Full-Text Search, а для аналітичних операцій – плагін MySQL Enterprise Edition, який надає розширені аналітичні функції.

Щоб спростити управління схемою бази даних, можна використовувати інструменти для моделювання даних, такі як MySQL Workbench. Вони дозволяють зручно додавати, змінювати або видаляти таблиці та структуру бази даних.

За браком вбудованої підтримки JSON в MySQL можна використовувати додаткові розширення, такі як MySQL JSON Data Type або MySQL JSON Functions. Вони додають підтримку роботи з JSON-даними, що дозволяє зручно працювати з цим форматом.

MongoDB – популярна документно-орієнтована база даних NoSQL, яка рекомендована під час розробки веб-додатків, які масштабуються та потребують великої бази даних для зберігання величезної кількості саме неструктурованих даних. MongoDB зберігає дані як документи у двійковому представленні, які називаються об'єктами BSON (Binary JSON), які є об'єктами, подібними до JSON у двійковому кодуванні. Пов'язана інформація зберігається разом для швидкого доступу до запитів через мову запитів MongoDB. Документи в MongoDB можна організувати в «колекції». Поля можуть відрізнятися від документа до документа; немає необхідності оголошувати системі структуру документів – документи описуються самі. Якщо до документа потрібно додати нове поле, це поле можна створити, не впливаючи на всі інші документи в колекції, не оновлюючи центральний системний каталог і не переводячи систему в автономний режим [5].

Переваги MongoDB:

– Масштабованість: MongoDB має високу масштабованість (горизонтально шляхом розподілу даних на кілька серверів (шардів)), що робить її придатною для програм із вимогами, що швидко змінюються, і великими наборами даних.

– Гнучкість: документоорієнтована модель даних MongoDB дозволяє розробникам зберігати та отримувати доступ до даних у більш гнучкий та інтуїтивно зрозумілий спосіб, ніж традиційні реляційні бази даних. Дана СУБД підтримує механізми реплікації, що дозволяють створювати резервні копії даних та забезпечувати високу доступність системи.

– Простота використання: MongoDB легко налаштувати та використовувати, завдяки простому та інтуїтивно зрозумілому інтерфейсу.

– Продуктивність: MongoDB відома своїми високопродуктивними можливостями з високою швидкістю читання та запису.

Недоліки MongoDB:

– Проблема узгодженості: гнучка модель даних MongoDB може ускладнити підтримку

узгодженості даних у різних документах і колекціях.

– Відсутність транзакцій: MongoDB не підтримує багатодокументні транзакції, що може бути недоліком для програм, які вимагають складних транзакцій.

– Використання пам'яті: використання пам'яті MongoDB може бути високим у порівнянні з іншими системами баз даних, особливо при роботі з великими наборами даних.

– Обмежена підтримка з'єднань: MongoDB не підтримує традиційні з'єднання у стилі SQL, що може ускладнити виконання певних типів запитів.

Згідно досліджень, збільшення робочого навантаження базується на збільшенні кількості підключених датчиків на станцію з 1 до 12 датчиків і, отже, збільшення кількості операцій вставки. MySQL приймає від 4000 записів при першій операції вставки до 48000 записів в останніх операціях вставки. З іншого боку, MongoDB приймає 4000 записів у всіх випадках від першої до останньої операції. У такому випадку найбільша затримка за умови 12 сенсорів та 48000 записів буде сягати у MySQL 72000 мілісекунд, а у MongoDB при такій же кількості сенсорів та 4000 записів буде сягати 1623 мілісекунд [5].

MySQL виявився більш продуктивним при вставці невеликої кількості записів та під час закінчення кожного періоду стабільності (поетапне збільшення кількості датчиків від 1 до 12). Зі збільшенням кількості вставлених записів і в кінці кожного періоду стабільності (від сценарію з 6 до 7 та від 11 до 12 датчиків), MySQL перевершує MongoDB. Однак, результати показали, що збільшення робочого навантаження у разі використання MySQL призводить до значної втрати продуктивності, більшої, ніж у випадку використання MongoDB. Тим не менш, MySQL випереджає MongoDB по розміру файлу БД: 2500 КБ проти 6340 КБ при 48 тисячах записів.

Інформація щодо розміру файлу БД може бути корисною в задачах, коли необхідно використовувати мікроконтролер для операцій з БД. ESP32 має кілька сотень кілобайт внутрішньої оперативної пам'яті, розташованої на тому ж кристалі, що й інші компоненти чіпа. Для деяких цілей цього може бути недостатньо, тому ESP32 має можливість використовувати до 4 МБ віртуальних адрес для зовнішньої пам'яті PSRAM (псевдостатична RAM). RAM – це «швидка» оперативна пам'ять, а PSRAM повільніша, проте коли необхідно збирати дані з сенсорів у межах об'єкта досліджень, наприклад, поля, цього буде цілком достатньо.

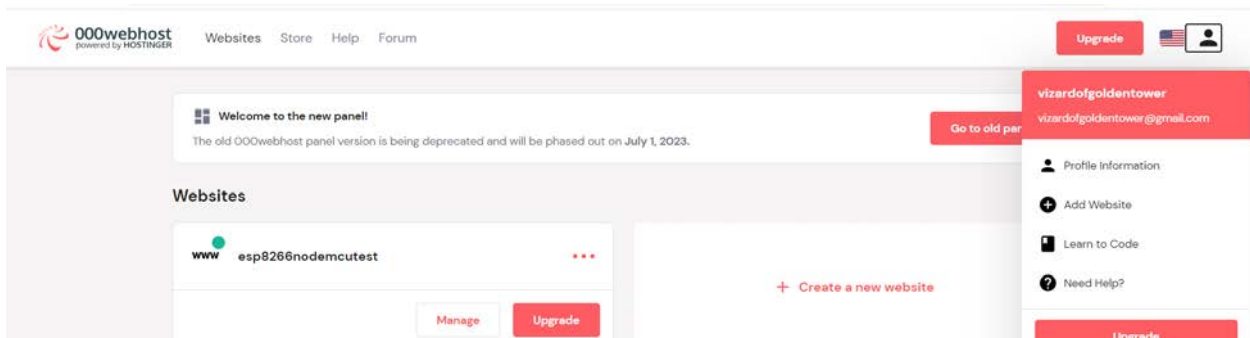


Рис. 1. Налаштований сервер

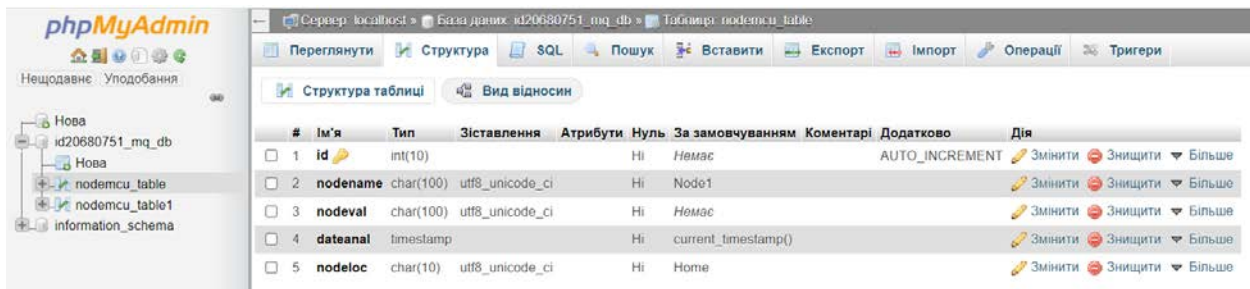


Рис. 2. Таблиці БД

Для реалізації підключення Esp8266 до MySQL було обрано наступний варіант ESP8266 → API сервер → MySQL. Плата ESP давала запит серверу на внесення даних, сервер перевіряв дозволи дані заносились до БД.

Для безпечної роботи необхідно було забезпечити плату достатнім рівнем шифрування і надсилати інформацію безпосередньо в БД або використовувати API сервер. У даному випадку був налаштований API сервер за допомогою PHP за адресою 000webhost.com.

Наступним кроком було створення БД та налаштування таблиць. Навіть без підключеного датчика інформація про пристрій стабільно надсилалась, що ви можете побачити на рисунках 3, 4 та 5. Для відображення інформації з БД на веб-сторінці був застосований спеціальний PHP скрипт.

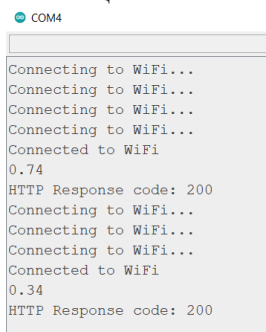


Рис. 3. Інформація з COM порту про успішну відправку даних

Висновки. Проведений аналіз реляційних та нереляційних баз даних у ході даного дослідження підкреслив важливі аспекти, які впливають на ефективне функціонування баз даних. Результати дослідження виявили, що серед розглянутих варіантів, MySQL виступає як оптимальний вибір для впровадження у розглянутий програмно-апаратний комплекс. MySQL виступає як ідеальний варіант за рахунок цінової політики Oracle, яка робить дану БД доступною та популярною в індустрії, яка забезпечує велику підтримку спільноти та наявність документації, що сприяє швидкому розв'язанню можливих проблем. Важливим фактором вибору MySQL є його здатність працювати на малопродуктивних серверах. Безпосередньо для нашого проекту MySQL компенсує відставання в швидкодії ефективністю управління розміром файлу БД, що дозволяє забезпечити високий ступінь оптимізації та продуктивності відносно інших варіантів. Однією з ключових переваг використання MySQL є його гнучкість та здатність до інтеграції з додатковим обладнанням. Встановлення розширеної пам'яті PSRAM відкриває шлях до більш ефективного опитування датчиків, забезпечуючи більш збалансований та продуктивний обмін даними у рамках розглянутого комплексу. Отже, обґрунтовано, що MySQL становить оптимальний вибір для впровадження у проект програмно-апаратного

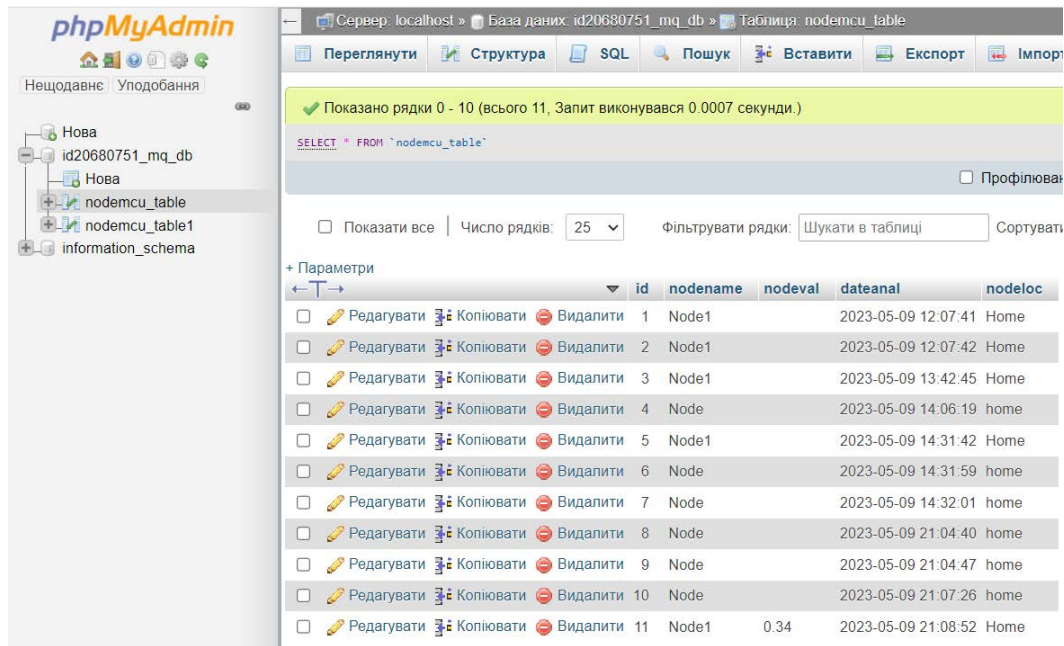


Рис. 4. Відображення інформації в MySQL

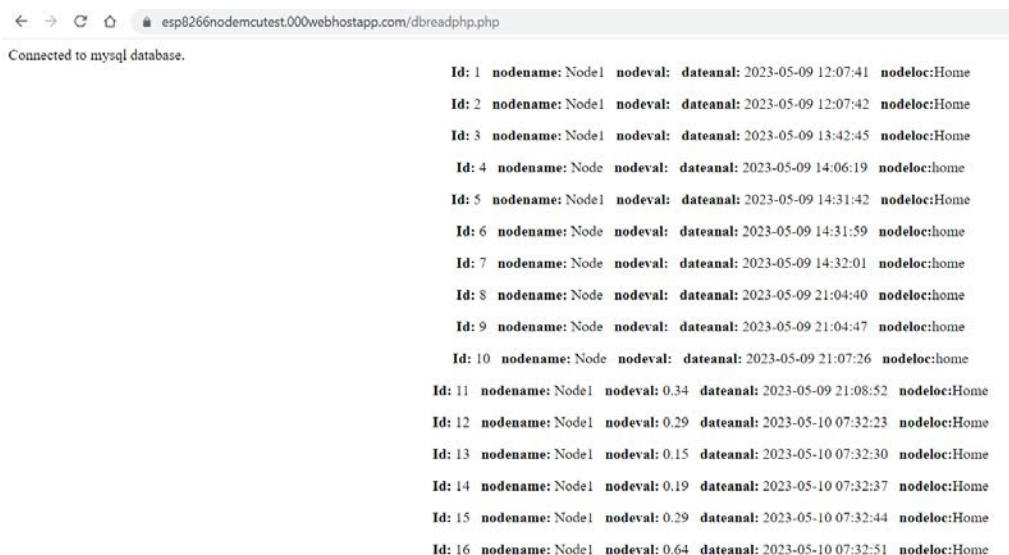


Рис. 5. Відображення інформації на сторінці

комплексу для виявлення небезпечних речовин. Його реалізація разом із розширеною пам'яттю PSRAM протягом дослідженого контексту може

дозволити досягти ефективного та надійного зберігання даних, забезпечуючи гармонійний розвиток проекту в майбутньому.

Список літератури:

1. Asabere, P., Sekyere, F., & Ofosu, W. K. (2020). Wireless biometric fingerprint attendance system using Arduino and MySQL database. *International Journal of Computer Science, Engineering and Applications (IJCSA)* Vol. 9.
2. Bell, C., & Bell, C. (2020). MySQL and Arduino: United at Last!. *Beginning Sensor Networks with XBee, Raspberry Pi, and Arduino: Sensing the World with Python and MicroPython*, 475-555.
3. Widagdo, D. Y. (2020). Sistem pencatatan hasil timbangan menggunakan sensor load cell melalui database berbasis arduino uno. *Journal of Telecommunication Network (Jurnal Jaringan Telekomunikasi)*, 10(1), 13-19.

4. Akbar, M. (2017). Realtime Database Sensor Menggunakan Arduino UNO Untuk Keperluan Sistem Informasi. *ILKOM Jurnal Ilmiah*, 9(1), 91-95.
5. Eyada, M. M., Saber, W., El Genidy, M. M., & Amer, F. (2020). Performance evaluation of IoT data management using MongoDB versus MySQL databases in different cloud environments. *IEEE access*, 8, 110656-110668.
6. Fernández De Villalta Canals, E. (2022). Data transmission via wifi and mobile data to a database using Arduino (Bachelor's thesis, Universitat Politècnica de Catalunya).
7. Elumalai, G., & Ramakrishnan, R. (2020). A novel approach to monitor and maintain database about physiological parameters of (Javelin) athletes using Internet of Things (IoT). *Wireless Personal Communications*, 111(1), 343-355.
8. Abdulkadhim, E. G. (2021, February). Design and develop an attendance system based on fingerprint and arduino board. In *Journal of Physics: Conference Series* (Vol. 1804, No. 1, p. 012011). IOP Publishing.
9. Ismail, N. S. N., Mustafa, S. Z. B., Yunus, F., & Abd Warif, N. B. (2020, June). Internet of Things (IoT) Smart Rubber Scale (SRS) System Using Arduino Platform. In *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)* (pp. 45-50). IEEE.
10. Arunachalam, A., & Andreasson, H. (2022). RaspberryPi-Arduino (RPA) powered smart mirrored and reconfigurable IoT facility for plant science research. *Internet Technology Letters*, 5(1), e272.
11. Li, C., & Gu, J. (2019). An integration approach of hybrid databases based on SQL in cloud computing environment. *Software: Practice and Experience*, 49(3), 401-422.

Karpenko M.I., Chumachenko S.M., Moshenskyi A.O., Guida O.G. CHOOSING THE OPTIMAL DATABASE FOR CREATING A SOFTWARE AND HARDWARE COMPLEX FOR THE DETECTION OF DANGEROUS SUBSTANCES

The article is devoted to the study of relational and non-relational databases in order to determine the optimal solution for integration into the future hardware and software complex designed for the detection of hazardous substances. The purpose of the study is to choose the best database (DB) for creating a hardware and software complex for detecting dangerous substances. To achieve this goal, various methods of analysis and synthesis were used: empirical, structural and genetic analysis, content analysis, experimental implementation, in particular, a literary analysis of scientific publications was conducted to study the features of relational and non-relational types of databases, as well as to highlight their main advantages and disadvantages. A selection of databases was selected for implementation in the project: MySQL, PostgreSQL, MongoDB, Redis and InfluxDB. The most optimal databases for the project were identified: MySQL and MongoDB. Weaknesses and strengths of each database have been identified. To demonstrate practical implementation, the ESP8266-based board was successfully connected to MySQL through the API interface of the "000webhost.com" service, demonstrating the possibility of integrating IoT applications. As a result, it was found that MySQL is the most optimal choice for implementation in this complex. This conclusion is based on several key aspects, including the affordability of MySQL thanks to Oracle's pricing policy, its popularity in the industry, significant support from the community, and the availability of documentation that facilitates the resolution of potential problems. A particularly important factor in the choice of MySQL was its ability to function effectively on low-power servers, which is essential for this project. Although MySQL is inferior to some other databases in terms of performance, its advantage lies in the ability to optimally manage the size of database files, which ensures a high level of performance. An additional advantage of MySQL is its flexibility and the ability to integrate with additional hardware, in particular with extended PSRAM memory, which facilitates more efficient polling of sensors and balanced data exchange. This allows for reliable and efficient data storage, which is critical for the successful development of the project in the future. Thus, MySQL, in combination with PSRAM, is the optimal solution for the implementation of a software-hardware complex focused on the detection of dangerous substances. The obtained results are of practical importance for the development of applications, hardware and software complexes, and programs that must interact with the database. This work can be useful for developers and specialists who plan to use the database in their projects.

Key words: MySQL, IoT, PSRAM, ESP8266, Arduino.