

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гудзевич А. В. Зелені «Оазиси» Вінниці. Вінниця: Логос, 2005. 32 с.
2. Вінниця: Генеральний план. Пояснювальна записка. Т. 1. Київ: Український державний НДІ проектування міст «Діпромісто», 2007. 146 с.
3. Як вінничани рятують парк «Хімік». *vn.depo.ua*. URL: <https://vn.depo.ua/ukr/vn/yak-vinnichani-ryatuyut-park-himik--12062016175600>
4. Вінниця. Дій, Замостя! Парк Хімік. *zamostia.vn.ua*. URL: <https://zamostia.vn.ua/lokatsiyi/park-himik/>
5. У парку «Хімік» почали зрізати дерева. *vn.depo.ua*. URL: <https://vn.depo.ua/ukr/vn/u-parku-khimik-pochalizrizati-dereva-202102231288183>
6. Різноманітні майданчики, прогулянкова алея та озеленення: як у Вінниці на Замості оновили парк «Хімік». *vezha.ua*. URL: <https://vezha.ua/riznomanitni-majdanchyky-progulyankova-aleya-ta-ozelenennya-yak-u-vinnytsi-na-zamosti-onovyly-park-himik/>
7. Дослідницький урбан-проект «Хімік». URL: [issuu.com.https://issuu.com/irm.vin/docs/khimik_about_process_](https://issuu.com/irm.vin/docs/khimik_about_process_)

УДК 004.031

ПОРІВНЯЛЬНИЙ АНАЛІЗ БАЗ ДАНИХ SQL ТА NOSQL

М. Р. Левченко, В. Ф. Гавліцький, О. В. Зелінська

Анотація. Проведено порівняльний аналіз двох основних типів баз даних: реляційних SQL і нереляційних NoSQL. Розглянуто їх природу, характеристики та застосування. Стаття висвітлює різницю у структурі даних, мовах запитів, схемах, масштабованості та інших аспектах. Порівняння допомагає визначити, який тип бази даних краще відповідає конкретним потребам в обробці та зберіганні даних.

Ключові слова: реляційні бази даних, нереляційні бази даних, SQL та NoSQL, структура даних, масштабованість даних.

У сучасному світі обробка та зберігання даних відіграє важливу роль у багатьох сферах, як-от у бізнесі, науці, технологіях та ін. Один із ключових виборів, який впливає на ефективність і надійність систем обробки даних – це вибір типу бази даних. У цій статті ми розглянемо та порівняємо два основні типи баз даних: реляційні SQL-бази даних та нереляційні NoSQL-бази даних.

Реляційна база даних (SQL) – цей тип баз даних є найдавнішим, його теоретичні основи були розроблені британським вченим Едгаром Коддом у 1970 році. Інформація у таких базах даних організована у вигляді таблиць, і кожна таблиця має визначену структуру з рядками (містять відомості про об'єкт – значення властивостей) та стовпцями (є властивостями цих об'єктів – полями). Кожен стовпець має встановлений тип даних, і дані зберігаються відповідно до цієї структури. Дані в реляційних базах даних зв'язані між собою, і доступ до них здійснюється за допомогою SQL-запитів [1].

Складні взаємозв'язки між об'єктами у реляційних базах даних моделюються за допомогою зовнішніх ключів, які вказують на інші таблиці. Це дає змогу підходити до проектування бази даних погляду нормалізації, що полягає у мінімізації дублювання інформації та зменшенні надмірності під час опису характеристик об'єктів [2].

Наприклад, якщо ми розглядаємо меню ресторану, то кожна страва має свої характеристики, як-от вага, ціна, назва, калорійність, а також категорія, до якої вона належить, як-от гарячі страви, холодні страви, перші страви, десерти, салати і под. Зв'язок між стравами та їх категоріями встановлюється за допомогою поля, яке містить індекс категорії в таблиці страв.

Таблиця 1

Страви

ID	Назва	Вага (г)	Ціна (грн)		
1	Борщ	300	60	250	3
2	Салат «Цезар»	200	80	350	5
3	Паста Болоньез	350	90	450	4
4	Чай чорний	0	20	0	6

У цьому прикладі таблиця «Страви» містить дані про страви: назви, вага, ціна, калорійність і посилання на категорію, до якої вони належать. Поле «Категорія ID» є зовнішнім ключем, який вказує на ідентифікатор категорії в таблиці «Категорії».

Таблиця 2

Категорії

ID	Назва
3	Гарячі страви
4	Перші страви
5	Салати
6	Напої

Це дає змогу легко класифікувати страви за категоріями і виконувати запити на кшталт «Покажи всі салати» або «Знайди всі гарячі страви з калорійністю менше 400». Зовнішні ключі допомагають створювати складні зв'язки між даними і зберігати їх у нормалізованому вигляді [2].

Нереляційна база даних (NoSQL) – це тип бази даних, де дані зберігаються без жорсткої структури та без чітких зв'язків між ними. Відмінність полягає у тому, що інформація може бути представлена у вигляді різноманітних документів, включаючи текст, зображення, відео та інше, і не вимагає строгого визначення структури даних. NoSQL-бази даних не використовують SQL для доступу до даних і надають більшу гнучкість у зберіганні та обробці нереляційних даних [1].

Усі переваги та недоліки реляційних баз даних базуються на жорсткій структуризації та типізації інформації про об'єкти. З одного боку, це дає змогу оптимізувати зберігання та індексування даних за допомогою нормалізації або денормалізації. З іншого боку, це ускладнює організацію зберігання та обробку погано структурованих (наприклад, дані об'єктів кешу) або зовсім не структурованих даних (наприклад, дані з різних джерел).

Для подолання цих обмежень було розроблено сімейство нереляційних баз даних [2].

Що таке SQL та NoSQL.

SQL – це мова програмування четвертого покоління з багатопарадигмальним характером. Вона поєднує в собі декларативну і процедурну парадигми. SQL був вперше випущений у 1986 році і став найбільш поширеною мовою для роботи з базами даних. SQL відповідає стандарту ISO/IEC 9075. Однак незважаючи на існування стандартів, SQL-код не завжди можливо перенести між різними системами управління базами даних без певних коригувань [3].

SQL складається з різних типів операторів для роботи з даними, які зберігаються в базах даних. Ці оператори можна поділити на такі категорії:

1. DDL (мова визначення даних). Ці запити використовуються для створення та модифікації схеми бази даних. Серед загальних команд DDL в SQL є CREATE (створення), ALTER (зміна) і DROP (видалення).

2. DML (мова обробки даних). Ці запити використовуються для виконання операцій з даними в базі даних, як-от вибірка, вставка, оновлення та видалення. Загальними командами DML в SQL є SELECT (вибірка), INSERT (вставка), UPDATE (оновлення) та DELETE (видалення).

3. DCL (мова контролю даних). Ці запити використовуються для контролю доступу до бази даних та надання прав доступу. Загальними командами DCL в SQL є GRANT (надання прав) та REVOKE (скасування прав).

4. TCL (Мова контролю транзакцій). Ці запити використовуються для управління транзакціями з метою забезпечення цілісності даних. Як загальні команди TCL у SQL включають BEGIN (початок транзакції), COMMIT (завершення транзакції) і ROLLBACK (відміна транзакції).

NoSQL (або Non-SQL, нереляційні бази даних). Це тип бази даних, який дає змогу зберігати та керувати даними, які мають нереляційну структуру, тобто не відображену в табличному форматі, і не мають табличних відносин.

NoSQL набирає все більшої популярності, особливо в додатках, які працюють з великими обсягами даних у режимі реального часу. Їх структури даних значно відрізняються від реляційних баз даних.

NoSQL є альтернативою традиційним реляційним базам даних, де дані зберігаються у вигляді таблиць, і структура даних передбачається перед створенням бази даних. Це особливо корисно для роботи з великими обсягами розподілених даних. NoSQL-бази даних мають масштабовану, високопродуктивну та гнучку природу.

Розрізняють чотири типи баз даних NoSQL.

1. **Стовпцеві (Columnar).** Бази даних стовпцевого типу зберігають та організують дані як стовпці, а не як рядки. Вони можуть ефективно запитувати великі обсяги даних, що розподілені в різних стовпцях. Це корисно для рекомендаційних систем, каталогів, виявлення шахрайства і под.

Приклади: Apache Cassandra, HBase, Google BigTable, Scylla, Vertica та ін.

2. **Документні (Document).** Бази даних документного типу зберігають напівструктуровані дані у форматі документів разом з їх описом. Кожен документ має унікальний ключ для адресації. Це корисно для управління вмістом і обробки даних у мобільних додатках. Вони часто використовуються разом із JSON та JavaScript.

Приклади: Apache CouchDB, MongoDB, MarkLogic, BaseX, IBM Domino та ін.

3. **Ключ-значення (Key-Value).** Бази даних ключ-значення використовують модель даних, основу на асоціативному масиві, де дані представлені парою ключ-значення. Вони корисні для управління сесіями і кешуванням у вебдодатках.

Приклади: Aerospike, Berkeley DB, Apache Cassandra, Redis, Riak, ZooKeeper та ін.

4. **Графові (Graph).** У графових сховищах дані представлені як вузли та ребра. Ця модель дає змогу представляти відносини між даними. Вони корисні для систем управління взаємозв'язками клієнтів, дорожніх карт, систем бронювання і под.

Приклади: AllegroGraph, InfiniteGraph, Neo4j, IBM Graph, Titan та ін. [3].

Таблиця 3

Порівняння SQL та NoSQL»[4–5]

Особливість	SQL (Реляційні бази даних)	NoSQL (Нереляційні бази даних)
Масштабованість	Вертикальна масштабованість	Горизонтальна масштабованість
Тип бази даних	Реляційні	Нереляційні або розподілені
Вік технології	Старі технології	Молода технологія
Структура даних	Таблична структура	Гнучка структура
Схема даних	Статична схема	Динамічна схема
Нормалізація	Підтримує нормалізацію	Підтримує денормалізацію
Запити	Підтримує складні запити	Має обмежені можливості для складних запитів
Мова запитів	SQL	У NoSQL немає стандартної мови запитів; синтаксис варіюється
Підтримка ACID	Підтримує ACID	Підтримує теорему CAP
Додавання нових даних	Вимагає змін у схемі	Дає змогу додавати дані без змін у схемі
Вартість	Дорожчий масштаб	Дешевший масштаб
Підтримка спільноти та постачальників	Широка підтримка	Обмежена підтримка спільноти
Застосування	Високорівневі транзакції	Важкі транзакції не є основним використанням
Ієрархічне зберігання даних	Не найкращий варіант	Добре підходить для ієрархічного зберігання

У дослідженні було розглянуто два основні типи баз даних: реляційні SQL-бази даних та нереляційні NoSQL-бази даних. Реляційні бази даних характеризуються жорсткою структуризацією, використанням SQL-запитів та можливістю гарної нормалізації даних. З іншого боку, NoSQL-бази даних надають гнучкість у зберіганні та обробці нереляційних даних, використовуючи різні структури даних. Вибір між цими двома типами баз даних залежить від конкретних вимог проекту [6].

Подальші дослідження в цьому напрямі можуть включати більш детальний аналіз конкретних властивостей різних систем баз даних, розгляд можливих варіантів їх комбінування та розробку оптимальних стратегій для використання SQL та NoSQL разом в одному проєкті. Також важливим напрямом може бути дослідження засобів синхронізації та реплікації даних між цими двома типами баз даних для забезпечення надійності та доступності інформації.

Abstract. This article provides a comparative analysis of two main types of databases: relational SQL and non-relational NoSQL. It explores their nature, characteristics, and applications. The article highlights the differences in data structure, query languages, schemas, scalability, and other aspects. The comparison helps determine which type of database best suits specific data processing and storage needs.

Keywords: Relational, Non-relational, SQL, NoSQL, Data structure, Scalability.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sem V. Альтернативна наука: SQL чи NoSQL – ось в чому питання. URL: <https://alternativescience.net/programming/242-sql-chy-nosql-os-v-chomu-pytannya/> (дата звернення 06.10.2023 р.).
2. Dou: Типи баз даних: особливості, відмінності та приклади. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення 06.10.2023 р.).
3. Що таке SQL. URL: https://uk.myservername.com/sql-vs-nosql-exact-differences#What_is_SQL (дата звернення 06.10.2023 р.).
4. Lauren Schaefer. MongoDB: NoSQL vs SQL Database. URL: <https://www.mongodb.com/nosql-explained/nosql-vs-sql> (дата звернення 06.10.2023 р.).
5. Різниця між SQL vs MySQL vs SQL Server. URL: <https://uk.myservername.com/difference-between-sql-vs-mysql-vs-sql-server> (дата звернення 06.10.2023 р.).
6. Огороднік М. О., Зелінська О. В. Переваги та недоліки реляційних та нереляційних баз даних. Прикладні аспекти сучасних міждисциплінарних досліджень: матеріали I Всеукраїнської науково-практичної конференції (м. Вінниця, 26 листопада 2021 р.). Вінниця: ДонНУ імені Василя Стуса. 2021. С. 106–109. URL: <https://jpasmd.donnu.edu.ua/issue/view/403> (дата звернення 06.10.2023 р.).

УДК 004.031.43:336.743-028.63]-047.44

РОЗРОБКА МЕТОДИКИ КОНТРОЛЮ ЗА ОБІГОМ ЦИФРОВИХ ФІНАНСОВИХ АКТИВІВ НА ОСНОВІ АНАЛІЗУ КРИПТОВАЛЮТНИХ ТРАНЗАКЦІЙ

Ю. О. Мазур, Л. В. Загоруйко

Анотація. У статті викладені методики контролю за обігом цифрових фінансових активів на основі аналізу криптовалютних транзакцій з використанням методів машинного навчання. Криптовалюти стали невід’ємною частиною сучасної фінансової системи, і їх обіг постійно зростає. Щоправда, відсутність ефективних засобів контролю та моніторингу може створити ризики для фінансової стабільності та безпеки. У статті пропонується новий підхід до контролю за обігом криптовалют, який використовує алгоритми машинного навчання для аналізу транзакцій та виявлення потенційних порушень.

Ключові слова: криптовалюта, транзакція, машинне навчання, ризики безпеки.

За останні роки цифрові фінансові активи, відомі також як криптовалюти, стали невід’ємною частиною сучасного фінансового середовища. Їх продажі та визнання з боку світової спільноти зростає, і це вже впливає на різні аспекти світових фінансів. Щоправда, зі зростанням популярності цифрових валют також зростає потенційна небезпека їх використання для незаконних операцій, включно з відмиванням грошей, фінансуванням тероризму та іншою злочинною діяльністю. Для забезпечення фінансової стабільності та безпеки необхідно розробити ефективні інструменти для контролю та моніторингу обігу цих цифрових активів. Однією з перспективних стратегій є використання аналізу криптовалютних транзакцій на основі методів машинного навчання. Це дає змогу автоматизувати процес виявлення потенційних порушень і незвичайних дій у світі криптовалют. Аналіз криптовалютних транзакцій може бути важливим інструментом для отримання інформації, яка сприяє виявленню транзакцій, які порушують закони або суперечать офіційній політиці держави в галузі криптовалют [1]. Цей вид аналізу може бути корисним для урядів і відповідних державних органів для виявлення зв’язків між підозрілими транзакціями та конкретними особами, а також для розкриття незаконних схем обігу та використання цифрових фінансових активів та криптовалют. Офіційні особи використовують різні методи та технології для виявлення осіб, які здійснюють незаконні криптовалютні транзакції.