ВІСНИК ПРИАЗОВСЬКОГО ДЕРЖАВНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
2025р. Серія: Технічні науки Вип. 51
p-ISSN: 2225-6733; e-ISSN: 2519-271X

УДК 004.65:004.45 DOI: 10.31498/2225-6733.51.2025.344596

ОГЛЯД БАЗ ДАНИХ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РІЗНИХ ОПЕРАЦІЙНИХ СИСТЕМ

Воротнікова З.Є. *канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: https://orcid.org/0000-0002-0746-5981, e-mail: vorotnikova_z_j@pstu.edu*

Сучасний світ баз даних характеризується постійними змінами, спричиненими появою нових технологій та зростанням потреб бізнесу. Серед ключових тенденцій варто виділити: зростання популярності хмарних баз даних, які забезпечують масштабованість та економію; активне використання NoSQL баз даних, що відрізняються гнучкістю; інтеграцію штучного інтелекту та машинного навчання для автоматизації процесів; поширення розподілених баз даних для обробки великих обсягів даних; зростаючий попит на аналітику даних в реальному часі; розвиток технологій «розрахунків в оперативній пам'яті» In-Memory Databases (IMDB) та підвищену увагу до безпеки даних. Правильний вибір системи управління базами даних (СУБД) є ключовим для успішної роботи програмного забезпечення, оскільки він безпосередньо впливає на її продуктивність, масштабованість, безпеку та ефективність. Нехтування особливостями операційної системи може призвести до серйозних проблем у майбутньому. Різні ОС мають різні архітектурні особливості, що впливають на продуктивність баз даних. Деякі бази даних, наприклад, Microsoft SQL Server, оптимізовані для конкретних ОС. Це дослідження представляє огляд сучасних систем управління базами даних з метою обрання оптимальної системи управління базами даних для конкретної задачі. Дослідження включає аналіз останніх публікацій, які підтвердили, що обміркований вибір СУБД потребує аналізу за багатьма критеріями і не є тривіальною задачею. Огляд літератури також показав, що дослідження у цьому напрямку є актуальними, затребуваними та забезпечують зручність і доступність в виборі оптимального програмного забезпечення. Було вивчено особливості функціонування 11 найпопулярніших СУБД станом на 2024 рік, зроблено порівняльний аналіз та надано рекомендації по вдалим областям застосування. Запропонований огляд сприяє раціоналізації обрання прийнятної СУБД для потреб розв'язання сучасних завдань обробки даних.

Ключові слова: *СУБД, SQL, NoSQL, IMDB, реляційна база даних, нереляційна база даних, операційна система, CAP-теорема.*

Постановка проблеми

Реалії сучасного світу породжують найвищий рівень вимог до електронних платформ. Більшість додатків так чи інакше потребують зберігання даних, починаючи від інформації про користувачів та їх публікації, закінчуючи даними телеметрії, такими як файли журналів та статистика використання. Найбільш поширеним і зручним засобом для зберігання подібної інформації є база даних, до якої можна звертатися через систему управління базами даних (СУБД). СУБД – це програмне забезпечення, призначене для створення та управління базами даних. Воно забезпечує структуроване зберігання даних та надає механізми для їх обробки та вилучення. СУБД виступають у ролі посередника між користувачем і базою даних. Структура БД може відрізнятися залежно від реалізації, але, як правило, це або набір файлів, що зберігаються на жорсткому диску або твердотільних накопичувачах, на яких СУБД серіалізує інформацію, або набір структур даних, що зберігаються безпосередньо в оперативній пам'яті In-Memory Databases (IMDB) [1].

Розробка структури бази даних (БД) може здійснюватись двома основними способами: за допомогою вбудованих засобів самої системи управління базами даних (СУБД) або з використанням мови моделювання UML, що підтримується зовнішніми CASE-засобами.

Використання засобів СУБД: наприклад, при створенні реляційної бази даних MySQL, PostgreSQL або Oracle можна використовувати вбудовані інструменти для проєктування та генерації схеми.

Зазвичай передбачається створення таблиць, визначення ключів, зв'язків, обмежень та індексів безпосередньо в інтерфейсі чи через скрипти SQL.

Використання UML та CASE-засобів:

UML (Unified Modeling Language) надає графічні нотації для моделювання структур даних, що полегшує візуалізацію та аналіз архітектури бази.

CASE-засоби (Computer-Aided Software Engineering), такі як ERWin та Enterprise Architect, дозволяють створювати діаграми класів, ER-діаграми та інші моделі для проєктування структури бази даних [2].

Ці інструменти допомагають визначити сутність, атрибути, зв'язки та обмеження, а також автоматично генерувати SQL-код для створення бази.

Приклад використання CASE-засобів:

У Enterprise Architect можна створити ER-діаграму, що моделює таблиці та зв'язки, а потім експортувати схему у вигляді SQL-скриптів для реалізації у вибраній СУБД.

Такий підхід сприяє більш чіткому плануванню та документуванню структури бази даних, а також полегшує її підтримку та розширення.

В цілому, вибір методу залежить від вимог проєкту, складності бази даних та переваг розробника. Використання UML і CASE-засобів найчастіше на етапах

p-ISSN: 2225-6733; e-ISSN: 2519-271X

проєктування, оскільки забезпечує наочність і зручність моделювання.

Зараз існує більше 300 всіляких СУБД. Одними з найбільш високопродуктивних, популярних і активно підтримуваних СУБД є Oracle Database, MySQL, Microsoft SQL Server, PostgreSQL, Redis [3] (відповідно до рейтингу DB-Engines Ranking). Серед всього різноманіття СУБД можна виділити деякі відмінності кожної з таких параметрах, як тип реалізованої БД, специфіка (область) застосування, тип доступу до коду (ліцензія, open-source) та ін.

CAP-теорема, запропонована Еріком Брюером на початку 2000-х років, є основним принципом у галузі розподілених систем. Вона стверджує, що розподілена база даних не може одночасно забезпечувати всі три властивості [4]:

Узгодженість (Consistency) – всі вузли системи бачать однаковий стан даних у будь-який момент часу.

Доступність (Availability) – кожен запит отримує відповідь незалежно стану системи, навіть при збоях.

Стійкість до поділу (Partition Tolerance) – система продовжує працювати нормально навіть при поділі мережі або у відмові частини вузлів.

Основний висновок CAP-теореми полягає в тому, що за наявності мережевих поділів (розділених мереж) розробникам доводиться робити вибір між забезпеченням узгодженості та доступності. Деякі системи наголошують на узгодженість (наприклад, традиційні реляційні бази даних). Інші – доступність і стійкість до поділів (наприклад, системи NoSQL, такі як Cassandra або Riak). Треті намагаються знаходити баланс, надаючи налаштування для компромісу між цими характеристиками.

Таким чином, проєктувальники та розробники повинні враховувати особливості своїх додатків та вибирати відповідну систему, усвідомлюючи обмеження CAP-теореми.

Вибір системи управління базами даних є складним багатопараметричним завданням та одним із ключових етапів розробки додатків баз даних. Правильний вибір програмного продукту має враховувати як поточні, так і майбутні потреби організації, забезпечуючи масштабованість і гнучкість системи. При цьому необхідно враховувати фінансові витрати на придбання обладнання, системи та програмного забезпечення, а також витрати на навчання персоналу та впровадження.

Важливо провести комплексний аналіз, щоб визначити, яка СУБД найбільше відповідає специфіці бізнесу, обсягу даних, навантажень та вимог до безпеки. Також необхідно оцінити можливість інтеграції з існуючими системами та наявність підтримки з боку постачальника.

Вибрана система має не тільки відповідати технічним вимогам, але й забезпечувати реальні вигоди підприємству, такі як підвищення ефективності роботи, зниження витрат та покращення якості обслуговування клієнтів. Таким чином, ретельний аналіз та виважений підхід до вибору СУБД дозволяють забезпечити успішну реалізацію проєкту та довготривалу його ефективність. В умовах середовища, що постійно розривається, завдання вибору оптимальної СУБД як ніколи актуальне.

Аналіз останніх досліджень та публікацій

Існує значна кількість досліджень, що присвячені аналізу тенденцій у розвитку систем управління базами даних. Ці дослідження базуються на різноманітних класифікаціях, які дозволяють краще зрозуміти особливості та напрямки еволюції сучасних систем. У роботах [5, 6] представлено детальну класифікацію СУБД за кількома важливими ознаками. Зокрема, вони поділяють системи за ступенем універсальності: до загального призначення належать системи, здатні працювати з різноманітними типами даних і застосовуватися у різних галузях, тоді як спеціалізовані системи орієнтовані на конкретні завдання або сфери діяльності. Крім того, у класифікації враховується рівень продуктивності систем, що визначає їх ефективність у обробці великих обсягів даних та швидкодії. Ще одним важливим параметром є спосіб доступу до даних: системи можуть бути класифіковані за моделлю доступу – наприклад, реляційні, ієрархічні, мережеві або об'єктно-орієнтовані. Окрім цього, у дослідженнях [7, 8] розглядаються сучасні методи захисту даних у СУБД, що є критично важливим у контексті зростаючого обсягу інформації та підвищення вимог до її безпеки. Підсумовуючи, ці дослідження сприяють глибшому розумінню сучасних тенденцій розвитку систем управління базами даних і дозволяють визначити перспективні напрямки їх удосконалення та застосування у різних сферах.

У книгах [9, 10] розглядаються різні підходи та критерії, які можна застосовувати при виборі систем управління базами даних (СУБД). Наводяться вимоги до програмного забезпечення, що дозволяє більш ефективно та обґрунтовано підходити до процесу прийняття рішення. Зазвичай, найпростіший і найшвидший спосіб – це оцінка відповідності існуючих систем основним вимогам конкретного проєкту інформаційної системи. Такий підхід дозволяє швидко визначити найбільш підходящу СУБД без необхідності складних тестувань або додаткових витрат.

Проте існує й більш складний і дорогий спосіб – це створення прототипу або випробувального проєкту, що базується на кількох різних системах управління базами даних. Після проведення порівняльних тестів і аналізу результатів робиться вибір найбільш відповідної системи. Хоча цей метод і вимагає більших зусиль та ресурсів, він дозволяє отримати більш об’єктивну оцінку та врахувати специфічні особливості кожної системи перед остаточним вибором. Водночас, навіть при такому підході важливо обмежити коло кандидатів, спираючись на визначені критерії відбору, що допомагає зменшити кількість варіантів та зосередитися на найбільш перспективних.

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Критерії для вибору СУБД можна класифікувати на кілька груп:

– Моделювання даних – враховує архітектурні особливості системи та її функціональні можливості, зокрема підтримку різних типів даних і способів їх обробки [11, 12, 14].

– Контроль роботи системи – оцінка механізмів управління транзакціями, забезпечення цілісності та безпеки даних [11, 15].

– Особливості розробки програм – враховують зручність та ефективність процесу створення та підтримки додатків, що працюють з обраною СУБД [12].

– Продуктивність – швидкість обробки запитів, здатність працювати з великими обсягами даних і підтримувати високий рівень навантаження [12, 13, 15].

– Надійність – здатність системи стабільно функціонувати у різних умовах, механізми резервного копіювання та відновлення [11, 15].

– Вимоги до робочого середовища – сумісність із апаратним забезпеченням і операційною системою, а також з іншими використовуваними технологіями [14, 15].

Загалом, список вимог до СУБД, які використовуються під час аналізу та вибору, може значно варіюватися залежно від конкретних цілей проєкту, масштабів системи та специфіки задач, що вирішуються. Важливо враховувати всі ці фактори для прийняття обґрунтованого і оптимального рішення щодо вибору системи управління базами даних.

Проблемам узгодженості (consistency), доступності (availability) та стійкості до розділення (partition tolerance) присвячені роботи [16-18]. У статтях [16, 18] аналізується перехід від монолітних сховищ даних до архітектур на основі мікросервісів, відображається еволюція прийняття архітектурних рішень у розподілену епоху під призмою CAP-теореми. Робота [17] присвячена дослідженню компромісів, на які компанії повинні піти при поліпшенні узгодженості, доступності та стійкості до розділів за правилом Брюера. Стаття [19] присвячена пошукам компромісу між узгодженістю та доступністю при проєктуванні інформаційних систем. Особливостям реалізації рішень для Хмарних технологій присвячені роботи [19, 20]. У статті [20] розглядається критично важливий аспект узгодженості даних у глобально розподілених базах даних, зокрема, Azure Cosmos DB фірми Microsoft.

Порівнянням характеристик та можливостей різних СУБД присвячені роботи [21-23]. У статті [21] порівнюється агрегатні функції та функції Windows у трьох основних системах управління реляційними базами даних: Oracle, SQL Server та MySQL. Ці функції необхідні для обробки великого набору даних і підготовки його до ефективного аналізу. Крім того, у цій статті розглядається інтеграція можливостей машинного навчання та підключення до баз даних NoSQL у рамках цих платформ. Це дослідження покликане допомогти адміністраторам і розробникам баз даних вибрати найбільш відповідну СУБД відповідно до їх конкретних потреб у агрегації, машинному навчанні, інтеграції NoSQL. У роботі [22] порівнюються системи на основі пошуку, SQL та NoSQL. Метою даного дослідження є надання характеристик, класифікації та оцінки системи управління базами даних, яка називається NoSQL, в аналітиці великих даних. Численні дослідження, які порівнюють можливості баз даних SQL і NoSQL, зокрема Oracle RDBMS і NoSQL Document Database (MongoDB), з погляду масштабу, продуктивності, доступності, узгодженості та шардингу представлені в огляді [23].

Таким чином, наукова проблематика, пов’язана з вивченням та порівнянням можливостей та характеристик різних СУБД, є надзвичайно актуальною. Вона обумовлена потребою в консолідованій інформації про індустрію СУБД та досвід їх використання, що відповідає вимогам сучасного цифрового суспільства. Вирішення цієї проблеми дозволить підвищити інформованість та грамотність користувачів, оптимізувати їх витрати та забезпечити більш ефективну діяльність.

Мета статті

Метою роботи є висвітлення та порівняння основних характеристик сучасних популярних систем управління базами даних (СУБД) з метою вибору більш прийнятної СУБД для потреб розв’язання певних задач.

Виклад основного матеріалу

Зазвичай вибір бази даних зумовлений тим, що використовували раніше або застосували вже у суміжних проєктах. Але іноді для нового проекту є можливість вибрати щось нове. І правильний вибір бази даних стає одним із найважливіших рішень. Від цього залежить насамперед кількість зусиль, які потрібно докласти для вирішення того чи іншого завдання, терміни та бюджет проєкту, вартість фахівців, які володіють навичками роботи з тією чи іншою СУБД. Від бази даних великою мірою залежить продуктивність всього рішення та її надійність.

Незважаючи на те, що всі СУБД виконують спільну та важливу функцію забезпечення користувачам можливості створювати, редагувати та отримувати доступ до збереженої інформації, спосіб та методи роботи з даними, а також рівень автоматизації та зручності використання, можуть значно варіюватися між різними платформами. Крім того, функціональні можливості та набір інструментів, які пропонує кожна конкретна СУБД, часто різняться – одні системи мають більш розвинуті механізми безпеки, масштабування чи обробки даних, тоді як інші орієнтовані на більш вузький спектр завдань.

Що стосується документації та технічної підтримки, то вони також можуть бути представлені різним ступенем систематичності. Вартість підтримки та

p-ISSN: 2225-6733; e-ISSN: 2519-271X

обслуговування може значно коливатися залежно від виробника та рівня сервісу, що пропонується.

При виборі оптимальної бази даних для конкретної організації важливо враховувати не лише її технічні характеристики, але й зручність у користуванні, можливість масштабування під зростаючі потреби бізнесу, а також ступінь інтеграції з іншими використовуваними системами та програмними продуктами. Важливо переконатися, що обрана СУБД легко піддається інтеграції з вже існуючою інфраструктурою та може працювати в межах заданих технічних та бізнесових вимог.

Крім того, слід враховувати наскільки легко може адаптуватися та змінюватися ця система у міру зростання та змін організації. Врахування всіх цих аспектів дозволить обрати найбільш відповідний продукт, який не лише задовольнить поточні потреби, але й забезпечить гнучкість та довгострокову стабільність у майбутньому.

Однією з вимог до робочого середовища є перелік операційних систем, що підтримують роботу СУБД. Розробники систем управління базами даних стикаються з проблемою операційного середовища. Із цього приводу є крайні точки зору. Одні вважають, що СУБД повністю має залежати від операційної системи, інші, що ОС має надати необхідні і достатні засоби. Практично неможливо виготовити справді ефективну систему управління баз даних, не розширюючи можливості операційної системи.

Якби потрібно було розробити сервер баз даних, що працює один на виділеному комп'ютері, то найкращим рішенням було б створювати його в розрахунку на машину взагалі без операційної системи. Фактично, це означало б створення деякого гібрида ОС і СУБД [24], у якому не було б нічого зайвого, а необхідні кошти було б реалізовано оптимально. Але звичайною вимогою до серверів баз даних є те, що повинна існувати можливість їх запуску в суміші з іншими програмами користувача, які, природньо, не можуть виконуватися без підтримки операційної системи. Сучасні ОС дозволяють деяким програмам користувача обійти обмеження операційної системи, що заважають їм, що обумовило появу таких технології як in-memory, механізм ODBC (Open Database Connectivity) [25] та ін. Але все ж нехтування особливостями операційної системи може призвести до серйозних проблем у майбутньому. Різні ОС мають різні архітектурні рішення, що впливають на продуктивність баз даних. Деякі бази даних, наприклад, Microsoft SQL Server, взагалі оптимізовані для конкретних ОС.

Суттєвим моментом є вибір базової технології роботи з БД: класична реляційна SQL-база або NoSQL? У обох категорій є свої переваги та недоліки.

Реляційні бази даних (РБД) являють собою один з найпопулярніших і широко використовуваних типів систем для зберігання і обробки даних [26]. Вони організовують інформацію у вигляді таблиць, кожна з яких має чітку і строго визначену структуру. Така організація дозволяє легко працювати з даними, підтримувати їхню цілісність і забезпечувати швидкий доступ до необхідної інформації. Основною мовою для взаємодії з реляційними базами є SQL (Structured Query Language), яка використовується для формулювання запитів, управління даними, їхнього оновлення та контролю цілісності даних. У цих системах дані зберігаються у вигляді рядків (записів) і стовпців (атрибутів), що сприяє високому рівню структурованості та зручності у роботі з ними. Це дозволяє чітко визначати залежності між різними елементами даних і забезпечує ефективне їхнє використання.

Щодо масштабування таких систем, то найчастіше воно виконується вертикально, тобто за рахунок збільшення ресурсів одного сервера. Це може включати додавання процесорів, розширення оперативної пам'яті, збільшення обсягу дискового простору тощо. Такий підхід має свої недоліки, оскільки в процесі оновлень або міграцій сервер може припиняти роботу, що призводить до простоїв. Хоча горизонтальне масштабування (розподіл даних між кількома серверами) є можливим і дозволяє підвищити масштабованість системи, але воно вважається більш складним у реалізації і вимагає значних змін у структурі бази даних та додаткових зусиль для синхронізації та управління.

Реляційні системи демонструють високу продуктивність при обробці транзакцій з невеликими та середніми обсягами даних, особливо якщо використовуються індекси для прискорення пошуку і виконання з'єднань між таблицями. Проте з ростом обсягів даних або кількості одночасних запитів може спостерігатися зниження швидкості роботи системи, що вимагає додаткових оптимізацій.

Важливим аспектом роботи РБД є дотримання принципів ACID (Атомарність, Узгодженість, Ізоляція, Довговічність). Ці принципи забезпечують цілісність і надійність транзакцій, тому системи автоматично підтримують ці вимоги без необхідності додаткових зусиль з боку користувача або адміністратора. Це особливо важливо у сферах, де критично важливо зберігати точність і цілісність даних, наприклад, у фінансових додатках, системах бухгалтерського обліку або електронній комерції.

З іншого боку, нереляційні бази даних, або системи NoSQL [27], використовують альтернативні моделі зберігання даних: ключ-значення, документи у форматі JSON, графи та інші. Вони дозволяють зберігати неструктуровані або напівструктуровані дані, такі як тексти, зображення, мультимедійні файли. Запити в таких системах зазвичай є більш гнучкими і дозволяють працювати з даними у форматах, що не підходять для традиційних таблиць.

При зростанні обсягів даних і кількості запитів нереляційні системи зазвичай масштабуються горизонтально, шляхом додавання нових серверів у розподілену мережу. Це дозволяє зменшити навантаження на окремі вузли, підвищити доступність і забезпечити високу продуктивність роботи з великими обсягами

p-ISSN: 2225-6733; e-ISSN: 2519-271X

даних і численними користувачами. Однак системи NoSQL мають нижчий рівень безпеки і гарантій цілісності даних, що може бути недоліком у деяких сценаріях, де важлива абсолютна надійність.

Завдяки здатності зберігати різноманітні типи даних і легко масштабуватися у розподілених середовищах, популярність баз даних NoSQL швидко зростає. Вони особливо актуальні для швидкої розробки мінімально життєздатних продуктів (MVP) у стартапах і проєктах, що вимагають гнучкого підходу до структурування даних. Такі системи не потребують тривалого попереднього налаштування і дозволяють швидко вносити зміни у структуру даних без значних простоїв, що є важливою перевагою у сучасних умовах розвитку технологій.

Сучасні програмні системи все частіше використовують комбінацію реляційних та нереляційних баз даних (NewSQL), щоб отримати максимальну вигоду від їхніх сильних сторін [28]. Реляційні бази даних (SQL) чудово справляються зі складними запитами та транзакціями, тоді як нереляційні (NoSQL) забезпечують високу швидкість обробки великих обсягів даних. Реляційні бази даних особливо цінні для критично важливих даних, таких як фінансові операції, де цілісність інформації є першочерговою. Вони зараз активно розвиваються у напрямку долучення модулів, що підтримують функції NoSQL.

На рис. дано фрагмент діаграми популярності різних видів СУБД станом на 2024 р. [29]. Розглянемо найпопулярніші СУБД, проаналізуємо їх архітектурні принципи, оптимальні варіанти використання, а також плюси та мінуси кожної технології (табл. 1).

PostgreSQL є однією з провідних систем управління реляційними базами даних, яка займає почесне перше місце у світі серед інших популярних систем. Її популярність обумовлена високою гнучкістю, розширюваністю та відповідністю сучасним стандартам обробки даних [30].

PostgreSQL є об'єктно-реляційною системою управління базами даних, вона підтримує об'єктно-орієнтований підхід у межах реляційної моделі. Це дозволяє користувачам поєднувати табличні структури з об'єктно-орієнтованими концепціями для створення складних ієрархічних структур даних. Крім того, PostgreSQL спрямовано на зміцнення відповідності стандартам SQL та розширюваності системи, що забезпечує високий рівень сумісності та можливість адаптації під специфічні задачі.

Ця СУБД є кросплатформною, що сприяє широкому впровадженню у багатьох сферах та умовах експлуатації.

Завдяки широкому спектру функцій та можливостей автоматизації, PostgreSQL є ідеальним рішенням для аналізу і зберігання великих обсягів даних. Вона підходить для створення інструментів автоматизації баз даних, що потребують високої продуктивності, відповідності стандартам ACID та потужного SQL-движка. Це робить її популярною серед фінансових установ, телекомунікаційних компаній та організацій, що мають високі вимоги до обробки та збереження даних.

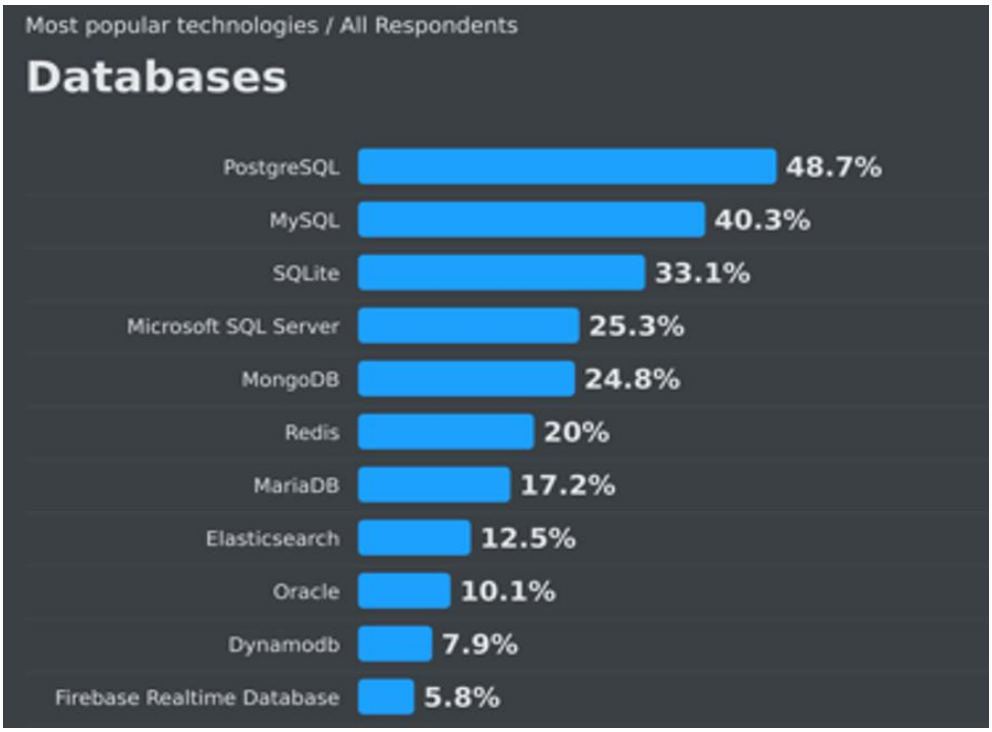

Рис. – Найпопулярніші СУБД у 2024 році (фрагмент діаграми)

p-ISSN: 2225-6733; e-ISSN: 2519-271X

Таблиця 1

Порівняння переваг та недоліків популярних СУБД

Система	Переваги	Недоліки
PostgreSQL	Надійність, масштабованість, підтримка JSON/XML/PostGIS, розширюваність, широкий спектр типів даних, багата екосистема, транзакції ACID, комерційні версії	Неповна документація, обмежені інструменти звітності та аудиту
MySQL	Безкоштовна простота, хороша інтеграція з PHP, підтримка хмар	Обмежена масштабованість, несумісність зі стандартами SQL, ліцензія Oracle
SQLite	Швидке читання, переносимість, легке встановлення, надійність, відсутність серверної архітектури	Обмежена підтримка типів та багатопоточності, відсутність серверної архітектури, не підтримує Unicode за замовчуванням
SQL Server	Простота управління, інтеграція з Microsoft, хороша документація, хмарні рішення	Висока ціна, тільки під Windows, складності при великих обсягах, потрібний кваліфікований персонал
MongoDB	Швидкодія, горизонтальне масштабування, гнучка інтеграція, зберігання в оперативній пам'яті	Високе споживання пам'яті, слабка реляційна підтримка, ризик витоків пам'яті, складність відновлення
Redis	Висока продуктивність, низька затримка, масштабованість, вбудований кеш	Залежність від оперативної пам’яті, обмежена функціональність запитів, підтримка транзакцій обмежена
MariaDB	Висока безпека, шифрування, нові функції (GIS, динамічні стовбці, користувацькі плагіни, механізми зберігання NoSQL, масштабованість), покращена продуктивність	Можливі проблеми зі стабільністю, молода спільнота, складнощі сумісності з MySQL
Elasticsearch	Надійна розподілена архітектура, хороша масштабованість, швидка обробка складних запитів	Підтримка лише JSON, обмежені інструменти моніторингу
Oracle 12c	Інноваційні технології, хмарна підтримка, обробка великих даних, висока надійність	Висока вартість, вимоги до потужної інфраструктури, складне освоєння для нових користувачів AWS
DynamoDB	Безсерверна, масштабована, проста налаштування, висока доступність	Обмежені можливості запитів, прив’язка до AWS, складнощі з аналітикою
Firebase	Простота для швидкого старту, легкий доступ і управління, хороша документація	Обмежені можливості складних запитів, складнощі при перенесенні даних

MySQL є однією з найпопулярніших СУБД у світі, яка з моменту свого створення пройшла значний шлях розвитку. Початково вона була випущена як рішення з відкритим вихідним кодом, що сприяло швидкому поширенню та широкому використанню у сфері веб-розробки. На теперішній час MySQL належить корпорації Oracle, що визначає її подальший розвиток та комерційну політику [31].

На сьогодні MySQL є основою класичних серверних веб-оточень та широко застосовується в популярних стекових рішеннях, таких як LAMP (Linux, Apache/nginx, MySQL, Perl/PHP/Python) та LEMP (Linux, nginx, MySQL, PHP/Python). Написаний мовами C і C++, MySQL відмінно підтримує роботу на різноманітних платформах: Linux, Windows, macOS, IRIX та інших, що забезпечує гнучкість і широкі можливості застосування.

MySQL є оптимальним рішенням для створення невисоко-навантажених веб-застосунків з обмеженим обсягом даних. Приклади таких застосувань включають локальні магазини електронної комерції, корпоративні сайти готелів або невеликі інтернет-портали. Оскільки системи OLAP (Online Analytical Processing) і OLTP (Online Transaction Processing) не вимагають складних запитів або обробки великих обсягів даних, MySQL є ідеальним інструментом у таких умовах. Крім того, варто розглянути використання MySQL у розробці інструментів бізнес-аналітики, де необхідна швидка обробка та аналіз даних.

SQLite – це швидка і легка однофайлова вбудована СУБД на мові C, яка не має сервера і дозволяє зберігати всю базу локально на одному пристрої. Для роботи SQLite не потрібні сторонні бібліотеки або служби. Поняття «вбудований» означає, що СУБД не використовує парадигму клієнт-сервер. Двигун SQLite –

це не окремий процес, з яким взаємодіє програма, а бібліотека. Програма компонується з нею, і двигун служить складовою програми. Як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Як і будь-яка СУБД, SQLite дозволяє записувати нову та запитувати існуючу інформацію, змінювати її, налаштовувати доступ. СУБД вбудовується у додаток та працює як його складова частина. Формат бази складається з одного текстового файлу, який можна прочитати на будь-якій платформі. Такий підхід підвищує продуктивність та швидкість роботи [32].

Працювати з SQLite можна як із бібліотекою, так і через SQLite3. SQLite3 – це консольна утиліта для роботи з SQLite від розробників СУБД. Вона запускається і працює у командному рядку, у консолі операційної системи. Ви можете завантажити версії для Windows, Mac OS та Linux.

За функціональністю SQLite3 є програмою-клієнт для клієнт-серверних додатків. З її допомогою можна вводити та передавати запити до бази даних: створювати, модифікувати, отримувати чи видаляти таблицю. Різниця в тому, що вона звертається не до окремого процесу-сервера, а до вбудованого в додаток движка SQLite.

Завдяки своїм властивостям SQLite застосовується на сайтах із низьким та середнім трафіком; у локальних однокористувацьких, мобільних додатках чи іграх, не призначених для масштабування; у програмах, які часто виконують прямі операції читання/запису на диск; у програмах для тестування бізнес-логіки. SQLite не вимагає адміністрування та працює на мобільних пристроях, ігрових приставках, телевізорах, безпілотних літальних апаратах, камерах, автомобільних мультимедійних системах тощо. СУБД використовує безліч програм: Firefox, Chrome, Safari, Skype, XnView, AIMP, Dropbox, Viber та інші.

Майкрософт SQL Server (MS SQL Server) являє собою високоефективний комерційний інструмент для управління реляційними базами даних, що посідає одне з провідних місць серед сучасних СУБД. Ця система забезпечує надійне зберігання, обробку та зміну структурованих даних, що дозволяє ефективно реалізовувати різноманітні бізнес-застосунки та інформаційні системи [33].

MS SQL Server використовує мову Transact-SQL (T-SQL), яка є розширенням стандартного SQL і забезпечує широкі можливості для програмування, управління та обробки даних. Це дозволяє розробникам і адміністраторам створювати складні запити, процедурні скрипти та автоматизовані рішення для роботи з даними.

MS SQL Server є високорівневим рішенням, яке ідеально підходить для компаній із наявною підпискою на продукти Microsoft та інфраструктурою, орієнтованою на екосистему Microsoft. Його потужні функції, глибока інтеграція з хмарними сервісами та широкі можливості для бізнес-аналітики роблять його одним із найпопулярніших і найзатребуваніших інструментів у корпоративному сегменті. Однак важливо враховувати високі витрати, складність налаштування та обмеження щодо операційної платформи при виборі цієї системи управління базами даних.

MongoDB – це сучасна нереляційна СУБД, що є безкоштовною та відкритою для модифікацій платформою, з комерційною версією для корпоративних клієнтів. Вона орієнтована на зберігання та обробку даних у форматі JSON-подібних документів, що забезпечує гнучкість у роботі як із структурованими, так і з неструктурованими даними. Такий підхід дозволяє застосовувати MongoDB до широкого спектру додатків, включаючи системи, що працюють із різнорідними даними [34].

Зв’язок між клієнтськими програмами та базою даних здійснюється через спеціальні драйвери, що забезпечують зручний інтерфейс для виконання операцій читання, запису, оновлення та видалення даних. Внутрішній кеш системи дозволяє одночасну обробку кількох типів даних, що підсилює швидкодію та ефективність роботи.

Завдяки високій швидкості та масштабованості, MongoDB ідеально підходить для систем, що потребують обробки даних у реальному часі. До прикладу, вона широко використовується для створення каталогів продуктів, де необхідно зберігати велику кількість об’єктів з різноманітними атрибутами. Крім того, MongoDB є актуальним рішенням для аналітичних платформ, оскільки її швидкодія дозволяє у режимі реального часу відстежувати поведінку користувачів та аналізувати великі обсяги даних.

Redis – це СУБД з відкритим вихідним кодом, що належить до класу NoSQL. Вона широко використовується як кеш-пам’ять, що забезпечує надзвичайно швидкий доступ до даних, що є критично важливим у сучасних високонавантажених системах [35]. Redis організовує дані у форматі пар «ключ-значення» і підтримує кілька структурованих типів даних, зокрема списки, множини та хеш-таблиці. Однією з ключових особливостей є можливість виконання реплікації даних та підтримки транзакцій, що дозволяє забезпечити високу доступність та цілісність інформації. Внутрішньою особливістю є обробка команд у черзі, що забезпечує ефективність та швидкість виконання операцій.

Основною сферою застосування Redis є сценарії, що вимагають високошвидкісного доступу до даних. Це включає, зокрема:

– Кешування даних. Зменшення навантаження на основні бази даних шляхом збереження часто запитуваної інформації у Redis, що суттєво підвищує швидкодію систем.

– Обробка даних для Інтернету речей (IoT). Великі обсяги даних, отримані з пристроїв IoT, можуть бути попередньо оброблені у Redis перед збереженням у довгостроковому сховищі.

– Мікросервісна архітектура. У системах з масштабованою хостинг-інфраструктурою Redis може

p-ISSN: 2225-6733; e-ISSN: 2519-271X

використовуватися як швидкий буфер або обробник даних, що не потребують довготривалого збереження.

MariaDB є відкритим та безкоштовним відгалуженням системи управління базами даних MySQL, яке підтримується на комерційній основі. Історично розроблялася як повністю сумісна з MySQL система, що дозволяє легко інтегрувати її у вже існуючі інформаційні системи без необхідності значних модифікацій. Основною особливістю MariaDB є відкритий вихідний код, що поширюється відповідно до ліцензії GNU General Public License (GPL), а також збереження сумісності з командами, API й бібліотеками, характерними для MySQL [36].

Оскільки MariaDB є системою, що тісно пов'язана з MySQL, її сфера застосувань здебільшого збігається з такою у MySQL. Водночас, очікується, що через покращену продуктивність та масштабованість MariaDB зможе стати більш ефективним рішенням для великих та високонавантажених систем. Це робить її перспективною платформою для сучасних корпоративних застосувань, що вимагають високої швидкодії, безпеки та гнучкості.

Elasticsearch – це архітектура NoSQL, яка орієнтована на документи СУБД, в основі якої лежить повнотекстова пошукова система. Побудований на базі бібліотеки Apache Lucene, вона зберігає дані у вигляді файлу JSON, підтримує API RESTful та використовує потужний аналітичний механізм для швидшого вилучення даних. Будучи програмним забезпеченням з відкритим вихідним кодом, вона включає як безкоштовні, так і платні версії [37].

У разі необхідності здійснення повнотекстового пошуку з БД (наприклад, пошук продукції на ресурсі для електронної комерції), ElasticSearch буде хорошим рішенням. Вона здатна швидко здійснювати пошук за величезним масивом даних і має велику функціональність (наприклад, вміє здійснювати пошук за іменованими категоріями).

Oracle Database 12c є СУБД, розробленою та підтримуваною корпорацією Oracle Corporation. Вона відзначається високою гнучкістю та здатністю підтримувати різноманітні моделі даних у межах однієї бази даних, зокрема моделі документів, графів, реляційної моделі та моделі ключ-значення, що дозволяє ефективно обробляти різні типи даних і забезпечує високий рівень інтеграції та масштабованості системи. Останні версії Oracle 12c орієнтовані на хмарні обчислювальні платформи, що дозволяє користувачам розгортати бази даних у гібридних або публічних хмарах, а також забезпечує додаткові можливості для автоматизації та управління ресурсами [38].

Ліцензування Oracle Database Engine є повністю запатентованим, що означає відсутність відкритого доступу до вихідного коду. Доступні як безкоштовні, так і платні ліцензійні опції, що дозволяє обрати рішення відповідно до масштабів та бюджету організації. В межах системи використовуються додаткові мови програмування, зокрема процедурно-орієнтована мова PL/SQL, яка розширює функціональність стандартного SQL, а також мова Java, що забезпечує можливість розробки складних додатків і процедур безпосередньо у середовищі бази даних.

Система Oracle 12c є високоефективним і гнучким рішенням для застосувань, таких як онлайн-транзакційні обробки (OLTP), сховища даних (Data Warehouse) та змішані (OLTP і Data Warehouse) системи. Вона особливо підходить для організацій із великими об'ємами даних – мільярдами записів – та достатнім бюджетом для підтримки та експлуатації. В умовах сучасних вимог до безпеки, масштабованості та високої продуктивності, Oracle Hybrid Cloud пропонує надійне і технологічно передове рішення для корпоративного сегменту.

DynamoDB – це безсерверна база даних NoSQL, розроблена компанією Amazon для підтримки свого інтернет-магазину, що стала доступною для розробників у 2012 році на платформі хмарного хостингу Amazon Web Services (AWS). Як і у випадку з магазином Amazon, DynamoDB відповідає на запити без обчислювальних витрат, необхідних для виконання складних з'єднань в реляційній базі даних. Оптимізоване для масштабування, це сховище елементів «ключ-значення», де всі дані, що відносяться до певного ключа, зберігаються разом з ним [39].

DynamoDB можна використовувати з програмами, що працюють під управлінням будь-яких операційних систем, включаючи Linux, Windows, iOS, Android, Solaris, AIX та HP-UX, вона легко інтегрується з різними мовами програмування та фреймворками.

Клієнти зі сфери фінансового обслуговування вибирають DynamoDB через стійкість, безпеку, продуктивність і масштабованість. DynamoDB допомагає дотримуватись галузевих норм та стандартів, включаючи SOC 1/2/3, PCI, FINMA та ISO. Також DynamoDB підходить для розробки програмних додатків в масштабі Інтернету з підтримкою метаданих контенту користувача і кешу, що вимагають паралельного підключення мільйонів користувачів і відправлення мільйонів запитів в секунду. Створення сховищ для мультимедійних метаданих з DynamoDB дозволить збільшити пропускну здатність та можливості паралельного запуску робочих навантажень мультимедіа та сфери розваг, таких як потокова передача відео в реальному часі та інтерактивний контент, а також забезпечити низьку затримку завдяки реплікації у кількох регіонах AWS. Для роздрібної торгівлі добре підходять проєктні шаблони для розгортання кошиків інтернет-магазинів, сервісів робочих процесів, відстеження запасів та ведення профілів клієнтів. DynamoDB підтримує події з високим трафіком та у великому масштабі та може обробляти мільйони запитів на секунду. Часто DynamoDB вибирають для масштабування ігрових платформ.

СУБД Firebase, що належить Google, це серверна частина та служба, яка використовується для розробки веб- та мобільного програмного забезпечення. Взагалі,

p-ISSN: 2225-6733; e-ISSN: 2519-271X

існує два варіанти: база даних Firebase (забезпечує доступ в режимі реального часу до даних, що знаходяться на різних платформах) та хмарний магазин Firestore (забезпечує більшу масштабованість та складніші моделі даних). Таким чином, обидва рішення чудово вписуються в сценарій, коли потрібно мати справу з великою кількістю даних у режимі реального часу: зміни до БД витягуються в міру їх виникнення [40].

СУБД Firebase рекомендується використовувати, коли програмне забезпечення має справу з даними в реальному часі, які необхідно синхронізувати між різними браузерами та пристроями. Типовий вибір: програми для обміну повідомленнями, програми для соціальних мереж та ігрові програми.

У таблиці 2 наведено порівняльний аналіз деяких параметрів найпопулярніших СУБД на сучасний момент.

Таблиця 2

Порівняльний аналіз параметрів СУБД

№ п/п	СУБД	Тип	Розробник	Операційна система	Ліцензія	Вихідний код	Підтримка SQL	Мmасштабування	Підтримка
1	Oracle Database	Мульти-модельна	Oracle Corporation	Linux, Microsoft Windows, Oracle Solaris, IBM AIX, HP-UX	Комерційна	***	+	По вертикалі та горизонталі	Платна
2	MySQL	Реляційна	Oracle Corporation	Linux, Microsoft Windows, Oracle Solaris, macOS, FreeBSD	GNU GPL та комерційна	*	+	Обмежено	Платна
3	SQL Server	Реляційна	Microsoft	Linux, Microsoft Windows	Комерційна	***	+	По вертикалі та горизонталі	Безкоштовна
4	PostgreSQL	Об’єктно-реляційна	PostgreSQL Global Development Group	Linux, Microsoft Windows, Oracle Solaris, IBM AIX, macOS, HP-UX, QNX	Вільне та відкрите програмне забезпечення, дозвільна ліцензія	*	+	По вертикалі	Платна
5	MongoDB	Документо-орієнтовна	MongoDB Inc.	Linux, Microsoft Windows, Oracle Solaris, FreeBSD, macOS	GNU AGPL (СУБД) и Apache License (драйвери)	*	-	По горизонталі через шардинг, реплікацію	Платна
6	MariaDB	Реляційна	MariaDB Corporation Ab, MariaDB Foundation	Linux, Microsoft Windows, Oracle Solaris, FreeBSD, macOS	GNU GPL	*	+	По горизонталі, реплікація, кластеризація	Платна, велика спільнота
7	SQLite	Реляційна	SQLite Consortium	Windows, macOS, Linux, Android, iOS та інші.	public domain (вільна ліцензія)	*	+	Обмежено	Платна
8	Redis	In-memory сховище ключ-значення	Salvatore "Antirez" Sanfilippo.	Linux, Windows (через підміну або емуляцію), macOS, FreeBSD та ін.	BSD-ліцензія (переважно)	*	-	По горизонталі, реплікація.	Платна, велика спільнота
9	Elasticsearch	Документно-орієнтована пошукова і аналітична система	Elastic NV	Linux, Windows, macOS, Docker, Kubernetes	Відкрита ліцензія, є комерційні версії	**	+	По горизонталі, кластеризація	Платна, активна спільнота

Продовження таблиці 2

№ п/п	СУБД	Тип	Розробник	Операційна система	Ліцензія	Вихідний код	Підтримка	Масштабування	Підтримка
10	DynamoDB	Ключ-значення та документо-орієнтована модель	Amazon Web Services (AWS)	Веб-сервіс, хмарна платформа (не залежить від ОС клієнта)	Платна послуга за моделлю оплати за використанням (pay-as-you-go)	***	+	Автоматичне масштабування у відповідь на навантаження	Безкоштовне
11	Firebase	Cloud Platform / Backend-as-a-Service (BaaS)	Google	Android, iOS, Web, Unity	Є комерційна і безкоштовна моделі	***	-	Автоматичне масштабування	Безкоштовне

* - Відкритий код; ** - Частково відкритий код; *** – Закритий код

З представленого спектру СУБД можна обрати ту, яка повністю відповідатиме специфічним потребам будь-якого підприємства. Наявність різноманітних безкоштовних варіантів дозволяє малим компаніям знайти оптимальний інструмент для ефективного управління базами даних, що відповідає їхнім вимогам і бюджету. Водночас, для великих підприємств або тих, хто потребує більш розвинених функцій, існує широкий вибір комерційних СУБД з додатковими можливостями та розширеним функціоналом. Це забезпечує гнучкість у підборі рішення, яке максимально відповідатиме складності та масштабам бізнес-процесів.

Висновки

У ході дослідження було проаналізовано підходи до побудови СУБД, надано увагу CAP- теоремі та перспективам розвитку описаних технологій в напряму хмарних технологій. Заторкнуті питання взаємодії СУБД з ОС та технологіям in-memory та ODBC.

Дослідження включає аналіз останніх публікацій, які підтвердили, що обміркований вибір СУБД потребує аналізу за багатьма критеріями і є нетривіальною задачею. Огляд літератури також показав, що дослідження у цьому напрямку є актуальними, затребуваними та забезпечують зручність і доступність в виборі оптимального програмного забезпечення.

Для порівняльного аналізу розглянуть 11 найпопулярніших СУБД станом на 2024 рік, наведено їх особливості, переваги та недоліки, надано рекомендації для більш вдалого застосування.

Перелік використаних джерел

[1] Shah A., Bhatt N. A systematic review of in-memory database over multi-tenancy. *International Journal of Electrical and Computer Engineering (IJECE)*. 2024. Vol. 14, no. 2. Pp. 1720-1729. DOI: https://doi.org/10.11591/ijece.v14i2.pp1720-1729.

[2] Ізмайлова О. В. Проектування інформаційних систем: навч. посіб. Київ: КНУБА, 2022. 88 с.

[3] The most popular database management systems. URL: https://db-engines.com/ (дата звернення 15.01.2025).

[4] Quantifying and generalizing the CAP theorem / E.A. Lee et al. arXiv preprint. arXiv:2109.07771. 2021. DOI: https://doi.org/10.48550/arXiv.2109.07771.

[5] Iglesias M.J.F. Brief introduction to database systems. Technical Report. 2023. 22 p. DOI: https://doi.org/10.13140/RG.2.2.35951.30884.

[6] Meier A., Kaufmann M. SQL & NoSQL databases. Models, Languages, Consistency Options and Architectures for Big Data Management. Wiesbaden : Springer Fachmedien Wiesbaden, 2019. 229 p. DOI: https://doi.org/10.1007/978-3-658-24549-8.

[7] Kaufmann M., Meier A. Datenbanksicherheit. *SQL-& NoSQL-Datenbanken*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2023. Pp. 125-178. DOI: https://doi.org/10.1007/978-3-662-67092-7_4.

[8] Nosql: Future of bigdata analytics characteristics and comparison with RDBMS / M. Arshad et al. *The Effect of Information Technology on Business and Marketing Intelligence Systems*. Cham : Springer International Publishing, 2023. Pp. 1927-1951. DOI: https://doi.org/10.1007/978-3-031-12382-5_106.

[9] Tamer Özsu M., Valduriez P. Principles of distributed database systems. Englewood Cliffs : Prentice Hall, 1999. 674 p. DOI: https://doi.org/10.1007/978-3-030-26253-2.

[10] Santos V. Data Modeling for Connected Data--A systematic literature review. arXiv preprint. arXiv:2410.10081. 2024. DOI: https://doi.org/10.48550/arXiv.2410.10081.

[11] Mostafa S.A., AbuSalim S.W., Saringat M.Z. A Comparative Study of Data Management Systems. *Journal of Soft Computing and Data Mining*. 2020. Vol. 1, no. 1. Pp. 10-16. DOI: https://doi.org/10.30880/jscdm.2020.01.01.002.

[12] Sinha R. A Comparative Analysis on different aspects of Database Management System. *JASC: Journal of Applied Science and Computations*. 2019. No. 1076-

5131. Pp. 2650-2667. DOI: https://doi.org/10.13140/RG.2.2.20503.50083.

[13] Vágner A. How Do NoSQL Databases Handle Variety of Big Data? *Proceedings of Ninth International Congress on Information and Communication Technology*, London, United Kingdom, 19-22 February 2024. Vol. 2. Pp. 459-469. DOI: https://doi.org/10.1007/978-981-97-3556-3_37.

[14] Illian M., Zinda C., Schlangenotto D. Data Store Architectures: Balancing Functionality and Performance. *2024 IEEE International Conference on Industrial Technology (ICIT)*, Bristol, United Kingdom, 25-27 March 2024. Pp. 1-6. DOI: https://doi.org/10.1109/ICIT58233.2024.10540811.

[15] A comprehensive and systematic literature review on the big data management techniques in the internet of things / Naghib A., Jafari Navimipour N., Hosseinzadeh M., Sharifi A. *Wireless Networks*. 2023. Vol. 29. Pp. 1085-1144. DOI: https://doi.org/10.1007/s11276-022-03177-5.

[16] Fadelelmoula A.A. Exploiting Cloud Computing and Web Services to Achieve Data Consistency, Availability, and Partition Tolerance in the Large-Scale Pervasive Systems. *International Journal of Interactive Mobile Technologies*. 2021. Vol. 15, no. 15. Pp. 74-102. DOI: https://doi.org/10.3991/ijim.v15i15.22517.

[17] Cuares F., Teleron J.I. Harmony in Nodes: Exploring Efficiency and Resilience in Distributed Systems. *Engineering and Technology Journal*. 2024. Vol. 9, no. 5. Pp. 4127-4136. DOI: https://doi.org/10.47191/etj/v9i05.32.

[18] Consistency vs. availability in distributed real-time systems / E.A. Lee et al. arXiv preprint. arXiv:2301.08906. 2023. DOI: https://doi.org/10.48550/arXiv.2301.08906.

[19] Vikiru A., Muiruri M., Ateya I. An overview on cloud distributed databases for business environments. arXiv preprint. arXiv:2301.10673. 2023. DOI: https://doi.org/10.48550/arXiv.2301.10673.

[20] Miryala N.K. NoSQL Database Services in Cloud–Overview Study. *International Journal of Advanced Research in Computer and Communication Engineering*. 2024. Vol. 13, no. 5. Pp. 637-647. DOI: https://doi.org/10.17148/IJARCCE.2024.13597.

[21] Hussein D.L. Evaluating Aggregate Functions and Machine Learning Integration: A Comparative Analysis of Performance, Security, and NoSQL Connectivity in Oracle, SQL Server, and MySQL. *UHD Journal of Science and Technology*. 2024. Vol. 8, no. 2. Pp. 7-23. DOI: https://doi.org/10.21928/uhdjst.v8n2y2024.pp7-23.

[22] Chai S., Qin Z. A Case Study of Cloud Query Performance Comparison Between SQL and NoSQL Database. *2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Bangkok, Thailand, 09-12 December 2024. Pp. 716-721. DOI: https://doi.org/10.1109/WI-IAT62293.2024.00117.

[23] SQL and NoSQL database software architecture performance analysis and assessments – a systematic literature review / W. Khan et al. *Big Data and Cognitive Computing*. 2023. Vol. 7, no. 2. Article 97. DOI: https://doi.org/10.3390/bdcc7020097.

[24] DBOS: a DBMS-oriented operating system / A. Skiadopoulos et al. *Proceedings of the VLDB Endowment*. 2021. Vol. 15, no. 1. Pp. 21-30. DOI: https://doi.org/10.14778/3485450.3485454.

[25] Bansal N., Sachdeva S., Awasthi L.K. Database Migration Tools: From RDB to NoSQL Database. *New Trends in Intelligent Software Methodologies, Tools and Techniques*. 2022. Pp. 563-570. DOI: https://doi.org/10.3233/FAIA220285.

[26] Sciore E. Database Design and Implementation. 2-nd ed. Springer Cham, 2020. 458 p. DOI: https://doi.org/10.1007/978-3-030-33836-7.

[27] Tripathi N. NoSQL database education: A review of models, tools and teaching methods. *Journal of Systems and Software*. 2025. Vol. 226. Article 112391. DOI: https://doi.org/10.1016/j.jss.2025.112391.

[28] Valduriez P., Jimenez-Peris R., Özsu M.T. Distributed database systems: The case for NewSQL. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII*. 2021. Pp. 1-15. DOI: https://doi.org/10.1007/978-3-662-63519-3_1.

[29] Stack Overflow Developer Survey 2024. URL: https://insights.stackoverflow.com/survey/2024 (дата звернення 18.01.2025).

[30] The PostgreSQL Global Development Group. URL: https://www.postgresql.org/about/ (дата звернення 18.01.2025).

[31] The official website MySQL. URL: https://www.mysql.com/ (дата звернення 25.01.2025).

[32] SQLite Home Page. URL: https://sqlite.org/ (дата звернення 25.01.2025).

[33] MS SQL Server. URL: https://www.microsoft.com/en-us/sql-server (дата звернення 28.01.2025).

[34] MongoDB Home Page. URL: https://www.mongodb.com/products (дата звернення 28.01.2025).

[35] Introduction to Redis. URL: https://redis.io/about/ (дата звернення 30.01.2025).

[36] MariaDB Foundation. URL: https://mariadb.org/ (дата звернення 25.01.2025).

[37] Elasticsearch B.V. URL: https://www.elastic.co/elasticsearch (дата звернення 15.01.2025).

[38] Oracle Home Page. URL: https://www.oracle.com/ (дата звернення 26.01.2025).

[39] Amazon Web Services, Inc. and its affiliates. URL: https://aws.amazon.com/partners/mongodb/ (дата звернення 30.01.2025).

[40] Firebase Home Page. URL: https://firebase.google.com/ (дата звернення 30.01.2025).

p-ISSN: 2225-6733; e-ISSN: 2519-271X

DATABASE REVIEW FOR SOFTWARE DEVELOPMENT ACROSS DIFFERENT OPERATING SYSTEMS

Vorotnikova Z. *PhD (Engineering), associate professor, SHEI «Priazovskyi state technical university», Dnipro, ORCID: https://orcid.org/0000-0002-0746-5981, e-mail: vorotnikova_z_j@pstu.edu*

The modern world of databases is characterized by constant changes driven by the emergence of new technologies and the growing needs of businesses. Key trends include: the increasing popularity of cloud databases, which offer scalability and cost savings; active use of NoSQL databases, known for their flexibility; the integration of artificial intelligence and machine learning for process automation; the spread of distributed databases for handling large volumes of data; the rising demand for real-time data analytics; the development of In-Memory Databases (IMDB) technologies for fast computing; and heightened attention to data security. Choosing the correct Database Management System (DBMS) is crucial for the successful operation of software, as it directly impacts performance, scalability, security, and efficiency. Neglecting the features of the operating system can lead to serious problems in the future. Different operating systems have distinct architectural characteristics that influence database performance. For example, some databases, such as Microsoft SQL Server, are optimized for specific OS environments. This research presents an overview of modern database management systems with the aim of selecting the optimal DBMS for specific tasks. The study includes analysis of recent publications confirming that a well-considered choice of a DBMS requires multi-criteria analysis and is a non-trivial task. The literature review also demonstrates that research in this area remains relevant and in demand, providing convenience and accessibility in selecting the most suitable software. The features of 11 of the most popular DBMSs as of 2024 have been examined, a comparative analysis has been conducted, and recommendations for optimal application areas have been provided. The proposed overview facilitates the rational selection of an appropriate DBMS to meet the needs of modern data processing tasks.

Keywords: *DBMS, SQL, NoSQL, IMDB, relational database, non-relational database, operating system, CAP theorem.*

References

[1] A. Shah, and N. Bhatt, "A systematic review of in-memory database over multi-tenancy," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 2, pp. 1720-1729, 2024. **doi: 10.11591/ijece.v14i2.pp1720-1729**.

[2] O.V. Izmailova, *Proektuvannia informatsiinykh system: navch. posib.* [Designing information systems: a textbook]. Kyiv, Ukraine: KNUBA Publ., 2022. (Ukr.)

[3] The most popular database management systems. [Online]. Available: https://db-engines.com/. Accessed on: January 10, 2025.

[4] E. Lee, S. Bateni, S. Lin, M. Lohstroh, and C. Menard, "Quantifying and generalizing the CAP theorem," 2021, *arXiv:2109.07771*. **doi: 10.48550/arXiv.2109.07771**.

[5] M.J.F. Iglesias, "Brief introduction to database systems," Technical Report, 2023. **doi: 10.13140/RG.2.2.35951.30884**.

[6] A. Meier, and M. Kaufmann, *SQL & NoSQL databases. Models, Languages, Consistency Options and Architectures for Big Data Management*. Wiesbaden, Germany: Springer Fachmedien Wiesbaden, 2019. **doi: 10.1007/978-3-658-24549-8**.

[7] M. Kaufmann, and A. Meier, "Datenbanksicherheit," in *SQL-& NoSQL-Datenbanken*, Berlin, Heidelberg : Springer Berlin Heidelberg, 2023, pp. 125-178. **doi: 10.1007/978-3-662-67092-7_4**. (Ger.)

[8] M. Arshad, M. Brohi, T. Soomro, T. Ghazal, H. Alzoubi, and M. Alshurideh, "Nosql: Future of bigdata analytics characteristics and comparison with RDBMS," in *The Effect of Information Technology on Business and Marketing Intelligence Systems*, Cham : Springer International Publishing, 2023, pp. 1927-1951. **doi: 10.1007/978-3-031-12382-5_106**.

[9] M. Tamer Özsu, and P. Valduriez, *Principles of distributed database systems*. Englewood Cliffs : Prentice Hall, 1999. **doi: 10.1007/978-3-030-26253-2**.

[10] V. Santos, "Data Modeling for Connected Data--A systematic literature review," 2024, *arXiv:2410.10081*. **doi: 10.48550/arXiv.2410.10081**.

[11] S.A. Mostafa, S.W. AbuSalim, and M.Z. Saringat, "A Comparative Study of Data Management Systems," *Journal of Soft Computing and Data Mining*, vol. 1, no. 1, pp. 10-16, 2020. **doi: 10.30880/jscdm.2020.01.01.002**.

[12] R. Sinha, "A Comparative Analysis on different aspects of Database Management System," *JASC: Journal of Applied Science and Computations*, no. 1076-5131, pp. 2650-2667, 2019. **doi: 10.13140/RG.2.2.20503.50083**.

[13] A. Vágner, "How Do NoSQL Databases Handle Variety of Big Data?," *in Proc. of 9-th Int. Congress on Information and Communication Technology*, London, United Kingdom, February 19-22, 2024, vol. 2, pp. 459-469. **doi: 10.1007/978-981-97-3556-3_37**.

[14] M. Illian, C. Zinda, and D. Schlangenotto, "Data Store Architectures: Balancing Functionality and Performance," in *Proc. of the IEEE International Conference on Industrial Technology (ICIT)*, Bristol, United Kingdom, March 25-27, 2024, pp. 1-6. **doi: 10.1109/ICIT58233.2024.10540811**.

[15] A. Naghib, N. Jafari Navimipour, M. Hosseinzadeh, and A. Sharifi, "A comprehensive and systematic literature review on the big data management techniques in the internet of things," *Wireless Networks*, vol. 29, pp. 1085-1144, 2023. **doi: 10.1007/s11276-022-03177-5**.

[16] A.A. Fadelelmoula, "Exploiting Cloud Computing and Web Services to Achieve Data Consistency, Availability, and Partition Tolerance in the Large-Scale Pervasive Systems," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 15, pp. 74-102, 2021. **doi: 10.3991/ijim.v15i15.22517**.

[17] F. Cuares, and J.I. Teleron, "Harmony in Nodes: Exploring Efficiency and Resilience in Distributed Systems," *Engineering and Technology Journal*, vol. 9, no. 5, pp. 4127-4136, 2024. **doi: 10.47191/etj/v9i05.32**.

[18] E. Lee, R. Akella, S. Bateni, S. Lin, M. Lohstroh, and C. Menard, "Consistency vs. availability in distributed real-time systems," 2023, *arXiv:2301.08906*. 2023. **doi: 10.48550/arXiv.2301.08906**.

[19] A. Vikiru, M. Muiruri, and I. Ateya, "An overview on cloud distributed databases for business environments," 2023, arXiv:2301.10673. **doi: 10.48550/arXiv.2301.10673**.

[20] N.K. Miryala, "NoSQL Database Services in Cloud–Overview Study," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 13, no. 5, pp. 637-647, 2024. **doi: 10.17148/IJARCCE.2024.13597**.

[21] D.L. Hussein, "Evaluating Aggregate Functions and Machine Learning Integration: A Comparative Analysis of Performance, Security, and NoSQL Connectivity in Oracle, SQL Server, and MySQL," *UHD Journal of Science and Technology*, vol. 8, no. 2, pp. 7-23, 2024. **doi: 10.21928/uhdjst.v8n2y2024.pp7-23**.

[22] S. Chai, and Z. Qin, "A Case Study of Cloud Query Performance Comparison Between SQL and NoSQL Database," in *Proc. of the 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Bangkok, Thailand, December 09-12, 2024, pp. 716-721. **doi: 10.1109/WI-IAT62293.2024.00117**.

[23] W. Khan, T. Kumar, C. Zhang, K. Raj, A. Roy, and B. Luo, "SQL and NoSQL database software architecture performance analysis and assessments – a systematic literature review," *Big Data and Cognitive Computing*, vol. 7, no. 2, article 97, 2023. **doi: 10.3390/bdcc7020097**.

[24] A. Skiadopoulos et al., "DBOS: a DBMS-oriented operating system," in *Proc. of the VLDB Endowment*, 2021, vol. 15, no. 1, pp. 21-30. **doi: 10.14778/3485450.3485454**.

[25] N. Bansal, S. Sachdeva, and L.K. Awasthi, "Database Migration Tools: From RDB to NoSQL Database," in *New Trends in Intelligent Software Methodologies, Tools and Techniques*, 2022, pp. 563-570. **doi: 10.3233/FAIA220285**.

[26] E. Sciore, *Database Design and Implementation*, 2-nd ed. Springer Cham, 2020. **doi: 10.1007/978-3-030-33836-7**.

[27] N. Tripathi, "NoSQL database education: A review of models, tools and teaching methods," *Journal of Systems and Software*, vol. 226, article 112391, 2025. **doi: 10.1016/j.jss.2025.112391**.

[28] P. Valduriez, R. Jimenez-Peris, and M.T. Özsu, "Distributed database systems: The case for NewSQL," *Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII*, pp. 1-15, 2021. **doi: 10.1007/978-3-662-63519-3_1**.

[29] Stack Overflow Developer Survey 2024. [Online]. Available: https://insights.stackoverflow.com/survey/2024. Accessed on: January 18, 2025.

[30] The PostgreSQL Global Development Group. [Online]. Available: https://www.postgresql.org/about/. Accessed on: January 18, 2025.

[31] The official website MySQL. [Online]. Available: https://www.mysql.com/. Accessed on: January 25, 2025.

[32] SQLite Home Page. URL: https://sqlite.org/. Accessed on: January 25, 2025.

[33] MS SQL Server. [Online]. Available: https://www.microsoft.com/en-us/sql-server. Accessed on: January 28, 2025.

[34] MongoDB Home Page. [Online]. Available: https://www.mongodb.com/products. Accessed on: January 28, 2025.

[35] Introduction to Redis. [Online]. Available: https://redis.io/about/. Accessed on: January 30, 2025.

[36] MariaDB Foundation. URL: https://mariadb.org/. Accessed on: January 28, 2025.

[37] Elasticsearch B.V. [Online]. Available: https://www.elastic.co/elasticsearch. Accessed on: January 15, 2025.

[38] Oracle Home Page. URL: https://www.oracle.com/. Accessed on: January 26, 2025.

[39] Amazon Web Services, Inc. and its affiliates. [Online]. Available: https://aws.amazon.com/partners/mongodb/. Accessed on: January 30, 2025.

[40] Firebase Home Page. [Online]. Available: https://firebase.google.com/. Accessed on: January 30, 2025.

Стаття надійшла 16.08.2025
Стаття прийнята 01.07.2025
Стаття опублікована 30.10.2025

Цитуйте цю статтю як: Воротнікова З. Є. Огляд баз даних при розробці програмного забезпечення для різних операційних систем. *Вісник Приазовського державного технічного університету. Серія: Технічні науки*. 2025. Вип. 51. С. 19-31. DOI: https://doi.org/10.31498/2225-6733.51.2025.344596.