

<https://doi.org/10.31891/2219-9365-2026-86-27>

УДК 004.6:004.652:658

ЛИСАК Віктор

Хмельницький національний університет

<https://orcid.org/0000-0001-5352-7090>

email: lysak.viktor@khmmu.edu.ua

МИХАЛЬЧУК Ігор

Хмельницький національний університет

<https://orcid.org/0009-0002-5162-5986>

email: mykhalchukiv@khmmu.edu.ua

КРИТЕРІЇ ВИБОРУ СУБД ДЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАКЛАДУ ВИЩОЇ ОСВІТИ: ПОРІВНЯЛЬНИЙ АНАЛІЗ РІШЕНЬ З ВІДКРИТИМ КОДОМ

У статті здійснено комплексний порівняльний аналіз сучасних систем управління базами даних (СУБД) з відкритим кодом PostgreSQL, MySQL, SQLite та Firebird щодо їхньої придатності для впровадження в інформаційних системах закладів вищої освіти. Розглянуто основні функціональні, архітектурні, ліцензійні та виробничі характеристики, критерії безпеки, можливості масштабованості, вимоги до ресурсів, особливості підтримки стандартів SQL, механізми резервного копіювання і реплікації, а також рівень підтримки спільноти. Також увагу приділено практичному досвіду експлуатації університетських інформаційних систем. За підсумками аналізу наведено рекомендації щодо вибору оптимальної відкритої СУБД для використання у закладах вищої освіти з урахуванням економічних, організаційних та технічних аспектів.

Ключові слова: СУБД, відкрите програмне забезпечення, Firebird, PostgreSQL, MySQL, SQLite, інформаційна система, заклад вищої освіти, транзакції, масштабованість, міграція даних, автентифікація

LYSAK Viktor, MYKHALCHUK Ihor

Khmelnytskyi National University

CRITERIA FOR SELECTING A DBMS FOR A HIGHER EDUCATION INSTITUTION INFORMATION SYSTEM: A COMPARATIVE ANALYSIS OF OPEN-SOURCE SOLUTIONS

This article presents a comprehensive comparative analysis of modern open-source database management systems (DBMS), such as PostgreSQL, MySQL, SQLite, and Firebird, concerning their suitability for implementation in information systems of higher education institutions (HEIs). The study examines the core functional, architectural, licensing, and performance characteristics of each DBMS, as well as key aspects such as security, scalability, resource requirements, compliance with SQL standards, backup and replication mechanisms, and the level of community support. Particular attention is paid to practical experience deploying and maintaining university information systems, including the proprietary "Elektronnyi Universytet" (Electronic University) information system developed at Khmelnytskyi National University, which is built on open-source DBMSs.

The article explores the specifics of each DBMS in the context of higher education's unique needs and constraints, such as supporting diverse academic and administrative processes, managing large volumes of heterogeneous data, ensuring interoperability with e-learning platforms, and maintaining cost-effectiveness and long-term sustainability. The analysis addresses challenges HEIs face, including resource constraints, the migration from proprietary to open-source solutions, data security and compliance requirements, and the need for flexible, scalable, reliable systems.

Based on comparative criteria and practical implementation experience, the article provides clear recommendations for selecting the most appropriate open-source DBMS for various use cases within higher education institutions. These recommendations account for technical and functional features as well as economic, organizational, and human resource considerations and prospects for future system development and integration. The findings are relevant for IT leaders, system architects, and educational decision-makers seeking a balanced, evidence-based approach to database technology selection. The study underscores the importance of a systematic and context-aware evaluation and highlights the growing role of open-source solutions in the digital transformation of higher education.

Keywords: DBMS, open-source, Firebird, PostgreSQL, MySQL, SQLite, information system, higher education institution, transaction management, scalability, data migration, authentication, digital transformation

Стаття надійшла до редакції / Received 19.02.2026

Прийнята до друку / Accepted 09.04.2026

Опубліковано / Published 31.05.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© ЛИСАК Віктор, МИХАЛЬЧУК Ігор

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Стрімка цифровізація усіх сфер суспільства, зокрема освіти та науки, вимагає від закладів вищої освіти (ЗВО) впровадження сучасних інформаційних систем, які здатні забезпечити ефективне управління освітнім і науковим процесом. Такі інформаційні системи повинні бути не лише гнучкими до масштабування під впливом постійного зростання обсягів та складності даних, але й відповідати сучасним вимогам до безпеки, надійності, продуктивності та ефективності обробки інформації. Центральною складовою такої

інфраструктури є система управління базами даних (СУБД), від якості та властивостей якої безпосередньо залежить функціонування всієї ІТ-екосистеми освітнього закладу.

За останнє десятиріччя спостерігається тенденція до переходу від пропріетарних рішень до відкритих СУБД, що дозволяє знизити витрати на розробку та адміністрування, підвищити цифровий суверенітет і незалежність від комерційних постачальників програмного забезпечення. Для українських ЗВО це питання має не лише економічне, а й стратегічне значення, оскільки вибір платформи для створення або модернізації інформаційних систем впливає на автономність, швидкість впровадження інновацій та інтеграцію в міжнародний освітній простір.

Водночас, кожна із сучасних відкритих СУБД має свої архітектурні особливості, переваги та обмеження, що ускладнює процес прийняття рішення для вибору бази даних для власних потреб. З огляду на це, перед ЗВО постає науково-практичне завдання – здійснити обґрунтований вибір такої СУБД, яка найкраще відповідатиме реальним потребам, наявним ресурсам і можливим майбутнім викликам.

Актуальність даного дослідження також зумовлена тим, що інформаційні системи освітніх закладів часто інтегрують різні підсистеми (робочі місця викладачів і студентів, електронні бібліотеки, документообіг, модульне середовище для навчання, наукові бази тощо), а це вимагає від СУБД гнучкості, сумісності та підтримки різних сценаріїв навантаження на неї. У цьому контексті, питання вибору оптимальної відкритої СУБД для інформаційної системи ЗВО є як прикладною, так і науковою проблемою, вирішення якої має забезпечити не лише стабільність і ефективність роботи системи, а й її довгострокову стійкість до технологічних змін, кадрових ризиків, змін в освітньому процесі тощо.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Використання відкритих СУБД у закладах вищої освіти стало предметом активних дискусій серед вітчизняних та зарубіжних науковців і практиків. Багато ЗВО, зокрема й у Європі, впроваджують відкриті системи для забезпечення цифрового суверенітету та зниження витрат. Наприклад, національна освітня мережа Швейцарії Switch мігрувала з MySQL на MariaDB, підкреслюючи тим самим важливість прозорості та незалежності від комерційних постачальників [3]. Значна частина університетів США та Великої Британії використовує відкриті СУБД для організації власних інформаційних систем [1, 27].

Згідно з рейтингом DB-Engines від компанії Redgate найпопулярнішими відкритими СУБД є MySQL і PostgreSQL, причому остання демонструє стійке зростання популярності серед розробників і у 2024 році обігнала MySQL за даними Stack Overflow Survey [10, 28]. Водночас SQLite є найбільш поширеною вбудованою СУБД для локальних та мобільних застосунків [11]. Щодо Firebird, то вона також займає власну нішу та демонструє стабільну присутність у корпоративних та освітніх рішеннях [9, 10, 16].

Наукові праці вказують, що вибір тієї чи іншої СУБД залежить від характеру навантаження на систему, вимог до транзакцій, безпеки та особливостей її адміністрування. Зокрема, PostgreSQL рекомендується для вирішення складних аналітичних завдань, тоді як MySQL є оптимальним вибором для вебпроектів із високим навантаженням [11, 23]. Дослідження також вирізняють ефективне застосування СУБД Firebird у випадках, коли необхідно поєднати надійність роботи системи з мінімальним втручанням задля її адміністрування [2, 4]. Вітчизняні автори також аналізують успішний досвід впровадження відкритих СУБД у ЗВО (зокрема й у Хмельницькому національному університеті) та звертають увагу на економічні і організаційні переваги використання вільно поширюваного програмного забезпечення [2, 4, 5].

Окремої уваги заслуговують експериментальні дослідження продуктивності, які підтверджують переваги застосування PostgreSQL для складних запитів, а MySQL у сценаріях з інтенсивними операціями запису інформації [23, 28].

Таким чином, сучасна література підтверджує доцільність впровадження відкритих СУБД в інформаційних системах ЗВО, підкреслюючи необхідність зваженого підходу до вибору конкретної СУБД з урахуванням технічних, організаційних і кадрових факторів [2, 3, 26, 27, 28].

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Метою статті є узагальнення сучасного досвіду застосування відкритих СУБД в інформаційних системах та вироблення практичних рекомендацій щодо вибору оптимального рішення для ЗВО. Для цього проаналізовано функціональні можливості, переваги та функціональні обмеження найпоширеніших відкритих СУБД, а також практичний досвід їх впровадження в освітніх закладах. Реалізація поставленої мети дозволить сформулювати обґрунтовані критерії вибору СУБД з урахуванням потреб, ресурсів і перспектив розвитку ІТ-інфраструктури закладів освіти.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Інформаційна система Хмельницького національного університету «Електронний університет» вже багато років є ефективним засобом цифровізації управління освітньою діяльністю. Однією із основних її складових є Firebird, яку було обрано в якості центральної СУБД системи. Водночас, в ІТ-інфраструктурі

університету представлені й інші рішення – PostgreSQL, MySQL, MS SQL, Oracle, Redis, які забезпечують роботу різних підсистем.

Постійний розвиток апаратного і програмного забезпечення та зростання вимог до функціоналу інформаційних систем спонукають розробників до регулярного аналізу архітектури, оптимізації структури баз даних, оновлення програмного забезпечення та впровадження нових технологічних рішень. Це вимагає зваженого підходу до вибору СУБД із врахуванням актуальних потреб, наявних ресурсів, сучасних трендів та стратегічних планів.

Для обґрунтованого вибору оптимальної центральної СУБД було проведено порівняльний аналіз Firebird, PostgreSQL, MySQL і SQLite – найпопулярніших відкритих СУБД в освітньому та корпоративному середовищі. Кожна з них займає свою нішу на ринку, а розуміння їхніх відмінностей дозволяє підібрати оптимальне рішення під конкретні завдання.

Критерії, що використовувалися для оцінювання, були визначені нами з урахуванням практичних потреб нашого ЗВО: продуктивність, вимоги до апаратних ресурсів, тип ліцензування, підтримка SQL-стандартів, складність адміністрування, особливості резервування даних, масштабування, підтримка транзакцій тощо. Їхня порівняльна характеристика наведена в таблиці 1.

Архітектурні особливості та цільове призначення. Для коректного вибору оптимальної СУБД важливо враховувати їхні архітектурні особливості та початкове цільове призначення.

PostgreSQL спочатку розроблялася як академічний проект, орієнтований на підтримку повної реляційної моделі, розширюваність і стандартизацію. Її клієнт-серверна багатопотокова архітектура підтримує складні типи даних, надає можливість створення власних функцій і типів даних, а також забезпечує відповідність стандарту SQL:2016. Завдяки цьому PostgreSQL широко використовується для вирішення складних аналітичних завдань, а також у системах з високою транзакційною активністю [1, 11, 20].

MySQL була задумана як проста і швидка СУБД для вебдодатків, що визначило її фокусування на високій продуктивності під час типових транзакцій, простоті адміністрування та масштабованості для сценаріїв з великою кількістю операцій читання. Останні версії MySQL (8.0+) значно розширили підтримку SQL-стандарту і отримали нові можливості. Зважаючи на це система має переваги у простоті впровадження для електронної комерції та онлайн-сервісів [11, 18, 24].

СУБД SQLite проектувалася як вбудована файлова база даних для застосунків, яким потрібне локальне зберігання даних без серверної частини. Серед її важливих переваг є такі як: повна автономність, відсутність потреб в адмініструванні, мінімальні вимоги до ресурсів та портативність. SQLite ідеально підходить для мобільних пристроїв, настільних додатків, рішень для Інтернету речей і невеликих автономних систем [11, 25].

Серед відкритих СУБД Firebird займає унікальне місце. Її архітектура поєднує корпоративний рівень надійності, багатoversійну модель управління транзакціями (MVCC) та простоту розгортання. Firebird успадкувала переваги InterBase, а з появою версій 3+ в ній покращилась відкритість, було забезпечено кросплатформність та підтримку стандарту SQL. З виходом Firebird 5.0 система отримала базову підтримку JSON, мультитотокове резервне копіювання, профайлер запитів і додаткові оптимізації продуктивності, що робить її ще привабливішою для застосування у сферах де необхідний баланс між функціональністю, стабільністю та економічними витратами на впровадження [1, 2, 7, 8, 14].

Таким чином, кожна із розглянутих СУБД зайняла власну нішу у цифровій екосистемі: PostgreSQL – для аналітичних і транзакційних систем з високими вимогами до гнучкості та відповідності стандартам; MySQL – для вебдодатків і проектів з високим навантаженням, у яких важливими є простота адміністрування та можливості масштабування; SQLite використовується для автономних мобільних рішень з мінімальним супроводом; Firebird – для корпоративних, адміністративних інформаційних систем, яким необхідна надійність, простота і невисока вартість володіння [8, 10, 11, 14, 16].

Продуктивність та масштабованість. Аналіз продуктивності сучасних СУБД потребує врахування специфіки навантаження (читання, запис, аналітика, кількість одночасних з'єднань) та архітектури додатків.

PostgreSQL демонструє високу продуктивність у складних аналітичних запитах завдяки потужному оптимізатору, підтримці паралельного виконання запитів та різноманітних методів з'єднання таблиць. Система ефективно обробляє великі обсяги даних, що робить її привабливою для аналітичних систем, сховищ даних та університетських аналітичних підсистем [11, 20, 28]. Для масштабування PostgreSQL використовують потокову та логічну реплікацію, а також сторонні рішення для горизонтального масштабування [28].

СУБД MySQL з використанням ядра InnoDB забезпечує високу швидкість обробки транзакцій у сценаріях з великим навантаженням на читання, а тому вона є популярною серед розробників вебдодатків і електронної комерції [11, 18, 28]. MySQL оптимізована для швидкого виконання простих запитів, ефективно горизонтально масштабується через механізми реплікації, але у складних транзакційних сценаріях або при виконанні аналітичних запитів може поступатися PostgreSQL [11, 28].

Таблиця 1.

Порівняльний аналіз відкритих СУБД

	Назва СУБД			
	PostgreSQL	MySQL	SQLite	Firebird
Архітектура та призначення	Клієнт-серверна, багатопотокова	Клієнт-серверна, об'єктно-реляційна	Вбудована, файлова	Клієнт-серверна, вбудована, багатOVERсійна
Тип ліцензування	PostgreSQL License	GPL v2 / комерційна	Public Domain	IDPL/IPL
Критерій оцінювання	Оцінка за 5-ти бальною шкалою*			
	Коментар			
Продуктивність	5/5	4/5	3/5	4/5
	Висока для аналітичних запитів і великих обсягів вибірки	Висока у типових OLTP-запитах,	Висока швидкість для локальних операцій	Ефективна у транзакційних операціях середнього навантаження, збалансоване споживання ресурсів
Вимоги до апаратних ресурсів	2/5	3/5	5/5	4/5
	Оптимально 1–2 ГБ ОЗП; без достатніх ресурсів швидкість падає	InnoDB вимагає значний об'єм пам'яті для буферного пулу (70–80 %)	Практично немає, працює навіть на мобільних пристроях	Ефективно функціонує при 512 МБ–1 ГБ ОЗП, оптимальна для середовищ з обмеженими ресурсами
Підтримка SQL-стандартів	5/5	3/5	3/5	4/5
	Майже повна підтримка стандарту SQL:2016	Покращена у версії 8.0+, не повноцінна	Базовий SQL-92, прості JOIN-запити	Строга типізація, підтримка більшості функцій SQL:2016
Зручність адміністрування	2/5	3/5	5/5	5/5
	Вимагає досвідчених адміністраторів	Проста для базового використання	Практично не потребує адміністрування	«Розумні» стандартні налаштування, легке адміністрування
Розмір спільноти / екосистеми	5/5	4/5	1/5	2/5
	Широка спільнота, дуже багато розширень, хмарна підтримка	Широка спільнота, значна комерційна підтримка від Oracle	Мала спільнота, але стабільна підтримка розробників	Вузькоспеціалізована спільнота; невелика кількість плагінів та інструментів
Безпека та контроль доступу	5/5	4/5	2/5	4/5
	Підтримка RLS, SSL, LDAP/GSSAPI, система ролей і привілеїв	SSL, ролі, аутентифікація через плагіни; добре захищена	Безпека лише на рівні файлової системи	Ролі, права доступу, підтримка Windows/trusted аутентифікації
Резервне копіювання	5/5	5/5	1/5	5/5
	pg_dump, pg_basebackup, PITR, streaming-знімки, повна транзакційна цілісність	mysqldump, mysqlpump, MySQL Enterprise Backup	Копіювання файлу, VACUUM INTO, без гарантованої цілісності транзакцій	Інкрементне та повне резервне копіювання з транзакційною цілісністю через gbak/nbackup
Реплікація та відмовостійкість	5/5	5/5	0/5	3/5
	Потокова і логічна реплікація, «гаряче резервування», автовідновлення	Master–slave/master–master, кластер, MySQL Group Replication	Не підтримується	Тіньові копії, сторонні рішення для реплікації
Підтримка JSON/NoSQL	5/5	4/5	3/5	2/5
	Вбудована підтримка JSON/JSONB, індекси, оператори	JSON-стовпці, функції; X DevAPI.	JSON1-розширення для обмеженого CRUD-функціоналу	Базова підтримка JSON з версії 5.0
Тригери та збережені процедури	5/5	3/5	2/5	5/5
	Багато мов (PL/pgSQL, Python, Perl), event-тригери	SQL-процедури, обмежений функціонал тригерів	Простенькі тригери; збережених процедур і пакетів немає	Потужна PSQL, можливість створення пакетів, селективні тригери
Масштабованість	3/5	5/5	0/5	1/5
	Потребує сторонніх рішень	MySQL Cluster, шардинг, проксі	Не підтримується	Лише вертикальна (за рахунок ресурсів)
Підтримка транзакцій (ACID)	5/5	4/5	5/5	5/5
	Повна підтримка MVCC, всі рівні ізоляції	InnoDB – повна підтримка ACID, MyISAM – часткова	ACID підтримується з WAL-режимом	MVCC, snapshot isolation
Інструменти адміністрування	5/5	5/5	3/5	3/5
	pgAdmin, DBeaver, різноманітні CLI-засоби	MySQL Workbench, phpMyAdmin тощо	DB Browser for SQLite, CLI, DBeaver	FlameRobin, IBEExpert, DBeaver
Повнотекстовий пошук (FTS)	5/5	4/5	2/5	1/5
	Вбудований, GIN-індекси, багатомовний	MyISAM/InnoDB, boolean-режим	FTS3/FTS4/FTS5-розширення	Лише через зовнішні функції
Геопросторові можливості (GIS)	5/5	3/5	2/5	0/5
	PostGIS	Spatial extensions, OpenGIS-стандарти	SpatiaLite extension	Не підтримується, можливе лише через зовнішні засоби

*Примітка: Бальна оцінка відображає експертну думку авторів даного дослідження на основі аналізу наукових публікацій, документації та практичного досвіду впровадження

SQLite відзначається високою швидкістю локальних операцій завдяки відсутності мережевої взаємодії та мінімальним ресурсним витратам. Через це система широко використовується для мобільних і настільних додатків, але вона не розрахована на багатокористувацькі сценарії, інтенсивне навантаження чи горизонтальне масштабування [11, 25].

СУБД Firebird забезпечує гнучкий баланс між ефективністю транзакційної обробки та споживанням системних ресурсів. Система підтримує багатOVERсійну модель управління транзакціями (MVCC), що дозволяє обробляти як прості, так і складні запити із збереженням цілісності даних. Особливо добре Firebird проявляє себе у сценаріях з помірним і середнім навантаженням, де часто показує кращу продуктивність за MySQL і порівняно з PostgreSQL, споживаючи при цьому менше системних ресурсів [10, 14, 26]. Масштабування у Firebird переважно вертикальне; горизонтальна реплікація реалізується через сторонні рішення або спеціалізовані архітектурні підходи [14, 15].

Вимоги до системних ресурсів. Рівень ефективності використання системних ресурсів є надзвичайно важливим критерієм при виборі СУБД, особливо на хостингах або при потребі розгортання великої кількості екземплярів серверів баз даних.

СУБД PostgreSQL для оптимальної роботи вимагає значних апаратних ресурсів, насамперед оперативної пам'яті. Через те для вже запущених у виробничому режимі інсталяцій рекомендується 1–2 ГБ ОЗП як мінімум, а оптимальне налаштування буферного пулу часто передбачає виділення 25–40% загальної пам'яті сервера [11, 20, 23]. Проте висока функціональність та складна архітектура виправдовують ці вимоги для масштабних і аналітичних систем.

MySQL із ядром InnoDB також має суттєві вимоги до пам'яті. Для ефективної роботи буферного пулу InnoDB рекомендують виділяти 70–80% доступної ОЗП. Водночас, MySQL може запускатися й на менш потужному обладнанні, що дозволяє її використовувати у невеликих вебпроектах [11, 18].

СУБД SQLite вирізняється мінімальними вимогами до ресурсів, оскільки вона не має окремого серверного процесу. База даних існує як звичайний файл на диску, тому система може ефективно працювати навіть на мобільних пристроях, у вбудованих пристроях чи настільних додатках із малим обсягом оперативної пам'яті. Це забезпечує виняткову портативність і простоту розгортання, але водночас обмежує паралельність обробки запитів і масштабування [25, 26].

Firebird виділяється серед корпоративних СУБД ефективністю використання ресурсів. Система здатна ефективно функціонувати на серверах із 512 МБ – 1 ГБ ОЗП і забезпечувати при цьому функціонал корпоративного рівня. Для настільних або вбудованих застосунків можлива робота у режимі вбудованої бібліотеки, який ще менш вимогливий до ресурсів. Такий баланс між функціональністю і економічністю робить Firebird оптимальним вибором для побудови інформаційних систем у разі обмеженості фінансових бюджетів і потреб у масштабуванні [2, 14, 15, 22].

Ліцензійні моделі та правові аспекти. Ліцензійна політика СУБД суттєво впливає на стратегічні рішення щодо впровадження інформаційних систем, особливо в державному та освітньому секторі.

PostgreSQL розповсюджується під власною ліберальною PostgreSQL License, яка дозволяє безкоштовне використання, модифікацію та розповсюдження як у відкритих, так і в комерційних проєктах без обмежень чи додаткових фінансових витрат. Відсутність зобов'язань розкривати власні зміни у вихідному коді робить PostgreSQL максимально універсальною з правової точки зору [11, 20].

СУБД MySQL має подвійну ліцензійну модель: для відкритих проєктів застосовується GNU GPL v2, а для закритих / пропріетарних рішень потрібна окрема комерційна ліцензія від Oracle. Це означає, що для використання MySQL у закритих комерційних продуктах можуть виникати додаткові витрати, а також пов'язані з ліцензуванням юридичні обмеження [18, 26].

SQLite знаходиться у публічному домені. Це означає повну відсутність обмежень на використання, модифікацію та розповсюдження програмного забезпечення. Жодних ліцензійних чи фінансових зобов'язань, обов'язкових умов для розкриття змін чи реєстрації використання немає. За ступенем відкритості та простоти це найбільш ліберальний підхід серед популярних СУБД [25, 26].

СУБД Firebird поширюється за Initial Developer's Public License (IDPL) та InterBase Public License (IPL), які є різновидами вільних ліцензій. Вони дозволяють безкоштовне комерційне використання без ліцензійних виплат, але якщо користувач змінює код самої СУБД, то він зобов'язаний відкрити такі зміни для спільноти. Більшість проєктів на Firebird не модифікують вихідний код самої СУБД, тому таких обмежень не відчують. Таким чином, такий підхід робить Firebird привабливою для комерційних і освітніх проєктів [8, 14].

Підтримка стандартів SQL та сумісність. Рівень відповідності СУБД стандартам SQL є важливим фактором для розробників, оскільки впливає на переносимість SQL-коду між різними системами, сумісність з іншими рішеннями та складність реалізації кросплатформених інтеграцій.

PostgreSQL вважається однією з найбільш стандартизованих відкритих СУБД: система підтримує майже всі основні функції стандарту SQL:2016, зокрема виконні функції, рекурсивні запити, JSON-операції, аналітичні та агрегатні функції. Окрім цього, СУБД PostgreSQL пропонує такі унікальні розширення як масиви, користувацькі типи, можливість підключення сторонніх мов для процедур та розширення

функціональності через модулі. Такі опції роблять PostgreSQL привабливою для проєктів, де потрібна максимальна гнучкість у роботі з даними та розширюваність [11, 20, 28].

MySQL довгий час відставала від SQL-стандартів у питаннях строгості типізації, підтримки складних конструкцій та сумісності поведінки в особливих випадках. Однак з версії 8.0+ система суттєво розвинулася: були додані CTE-вирази, віконні функції, покращено роботу з JSON, однак ще залишаються певні особливості реалізації, що можуть вимагати певного доопрацювання коду при міграції чи інтеграції [18, 26].

СУБД SQLite реалізує основні можливості стандарту SQL-92, однак через динамічну типізацію даних та спрощену архітектуру не підтримує деякі такі оператори як RIGHT JOIN, FULL OUTER JOIN та обмежено реалізує складні функції. Для більшості простих і середніх застосунків цих можливостей достатньо, але при переносі великих систем або побудові складних звітів можуть виникати труднощі [25, 26].

Firebird демонструє високий рівень відповідності стандартам SQL, особливо починаючи з версії 3.0. Система підтримує віконні функції, CTE-вирази, розвинуту роботу з тригерами, пакетами, користувацькими функціями та більшість можливостей SQL:2016. Окремою перевагою Firebird є строга типізація та передбачувана поведінка у всіх SQL-операціях, що значно спрощує розробку складних бізнес-логік. Додатково СУБД Firebird пропонує власні розширення, зокрема удосконалені механізми для процедур і обробки подій [1, 14, 15].

Простота адміністрування та підтримки. Вартість та складність адміністрування є важливими чинниками для оцінювання довгострокових витрат на використання СУБД. Від цього залежить не лише супровід, а й кадрові вимоги та гнучкість розвитку інформаційної системи.

PostgreSQL має величезну кількість параметрів налаштування і вимагає від адміністраторів високої кваліфікації. Для досягнення максимальної продуктивності та стабільності потрібен досвід у адмініструванні, особливо при роботі зі складними запитами, налаштуванням реплікації чи кластеризації. Водночас, PostgreSQL має детальну документацію та велику спільноту, що значно полегшує пошук рішень і підтримку у складних випадках [20, 28].

СУБД MySQL відома своєю простотою у адмініструванні для базових сценаріїв. У більшості випадків розгортання та супровід здійснюється швидко, особливо завдяки великій кількості графічних інтерфейсів адміністрування. Однак у складних корпоративних середовищах налаштування реплікації, секціонування таблиць чи оптимізації продуктивності може вимагати додаткових знань у цій сфері [11, 18].

SQLite практично не потребує адміністрування у традиційному сенсі, оскільки є файловим «движком», який не має серверного процесу чи налаштувань мережевої взаємодії. Основні адміністративні дії обмежуються резервним копіюванням файлів бази даних і періодичною дефрагментацією. Все це позиціонує SQLite як одне із найзручніших рішень для невеликих автономних або вбудованих застосунків [11, 25].

СУБД Firebird пропонує потужний функціонал для якого необхідна відносно проста підтримка. Оптимальні налаштування за замовчуванням забезпечують стабільну роботу у більшості випадків, а основні адміністративні завдання (резервне копіювання, моніторинг, оптимізація) можуть виконуватися розробниками навіть без глибоких спеціалізованих знань у сфері адміністрування баз даних. На відміну від складніших корпоративних СУБД, Firebird має менше специфічних «особливостей», що знижує експлуатаційні ризики і полегшує впровадження в проєкти, коли вимоги до високої кваліфікації адміністраторів не критичні [1, 8, 14].

Екосистема та підтримка спільноти. Розвиток та стійкість відкритої СУБД значною мірою залежать від активності спільноти користувачів і розробників, кількості сторонніх інструментів, документації та навчальних ресурсів.

PostgreSQL має одну з найпотужніших і найактивніших спільнот серед усіх відкритих СУБД. Розробникам доступні тисячі розширень, а також зручні графічні і консольні інструменти адміністрування (pgAdmin, DBeaver, CLI тощо) та багато сучасних навчальних матеріалів. Окрім того, СУБД PostgreSQL повністю підтримується основними хмарними провайдерами (AWS, Azure, Google Cloud), що дозволяє інтегрувати її у складні багатохмарні архітектури [20, 28].

СУБД MySQL залишається дуже популярною завдяки значній кількості користувачів, широкому поширенню серед веброзробників, а також можливості отримати гарантовану платну підтримку від Oracle. Разом з тим, саме корпоративне управління з боку Oracle і викликає певні побоювання щодо майбутнього розвитку цього проєкту, а частина спільноти вже розпочала перехід на форки (MariaDB, Percona). Водночас, кількість сторонніх інструментів розробки та адміністрування, бібліотек розширень та навчальних курсів для MySQL на сьогодні залишається величезною [11, 18, 24].

SQLite розвивається завдяки невеликій, але висококваліфікованій команді. Екосистема тут не така багата на розширення чи інструменти, проте простота самої СУБД значно знижує потребу у складних засобах адміністрування та підтримки [25, 26].

СУБД Firebird має віддану та стабільну спільноту, хоча й меншу за чисельністю, порівняно з лідерами нашого аналізу. Розвиток системи відбувається передбачувано, з регулярними оновленнями, випуском повноцінної документації, організацією різноманітних конференцій та підтримкою через форуми і

спеціалізовані ресурси. Набір графічних та командних інструментів невеликий (FlameRobin, IBExpert, DBeaver), проте повністю закриває потреби для адміністрування та розробки, хоча екосистема розширень і партнерських рішень не така багата, як у PostgreSQL чи MySQL [8, 14, 15].

Засоби контролю доступу. Критерії організації безпеки та можливості управління доступом до даних є на теперішній час визначальними для використання СУБД у критичних системах, особливо при достатньо великій кількості користувачів.

PostgreSQL має одну з найбільш розвинених моделей організації безпеки серед відкритих СУБД: підтримує багаторівневу систему прав доступу (рольова модель), гнучку політику аутентифікації (включно з підтримкою SSL, LDAP, GSSAPI, Kerberos), а також функціонал захисту на рівня рядків (row-level security, RLS), який дозволяє обмежувати доступ до окремих рядків таблиць відповідно до політик безпеки. Ці можливості якраз і задовольняють вимоги до організації захисту для корпоративних політик та інформаційних систем органів державної влади [11, 20].

СУБД Firebird також забезпечує деталізоване управління правами доступу на рівні ролей, таблиць, полів та підтримує кілька методів аутентифікації, зокрема інтеграцію з Windows (trusted authentication). Останні версії розширили можливості контролю через підтримку зовнішніх плагінів і налаштовуваних політик. Все це дозволяє використовувати Firebird у бізнес-критичних системах із підвищеними вимогами до безпеки [8, 14].

СУБД MySQL надає базову рольову модель, засоби керування доступом до об'єктів бази, підтримку SSL та гнучкі механізми аутентифікації (у тому числі через плагіни). Проте для досягнення порівняного з корпоративними СУБД рівня захисту часто потрібні додаткові налаштування, а у великих системах ще й інтеграція із зовнішніми системами авторизації (LDAP, Kerberos) [1, 18].

SQLite не має внутрішніх засобів аутентифікації чи управління доступом на рівні самої бази. Контроль безпеки реалізується механізмами файлової системи операційної системи, що й обмежує можливості застосування цієї СУБД у багатокористувацьких системах, або IT-інфраструктурі з підвищеними вимогами до безпеки [11, 25].

Резервне копіювання та реплікація. Можливості резервного копіювання та реплікації визначають ступінь надійності та безперервності роботи інформаційних систем на базі конкретної СУБД.

СУБД Firebird забезпечує високу надійність резервного копіювання завдяки вбудованим засобам `gbak` і `pbackup`, які гарантують транзакційну цілісність даних навіть у процесі роботи системи. З версії 5.0 додано підтримку багатопотоковості, що підвищило ефективність резервування для великих баз даних [8, 14, 15]. Водночас можливості реплікації у Firebird залишаються доволі стандартними: для більшості випадків доступна лише організація тінювих копій або використання сторонніх рішень, тому ця СУБД краще підходить для систем малих і середніх організацій, які не вимагають складної розподіленої інфраструктури [7].

PostgreSQL пропонує розвинуті засоби резервного копіювання, серед яких є утиліти `pg_dump`, `pg_basebackup`, механізми Point-In-Time Recovery (PITR) та підтримка знімків файлової системи. Щодо реплікації, то PostgreSQL дозволяє використовувати як потокову, так і логічну реплікацію, гаряче резервування та механізми автоматичного оновлення у разі відмови. Це забезпечує високу доступність та масштабованість у корпоративних проєктах [20, 23].

СУБД MySQL також має сучасні інструменти для резервного копіювання: `mysqldump` та `mysqlpump` у відкритій ліцензії і MySQL Enterprise Backup у комерційних редакціях. Для масштабування і забезпечення безперервності роботи система пропонує широкий спектр схем реплікації: від master-slave/master-master до кластерних рішень (MySQL Group Replication, MySQL Cluster). Такий спектр інструментів резервування та реплікації робить СУБД MySQL затребуваною для масштабованих розподілених систем та вебпроєктів з високими вимогами до доступності [11, 18].

В свою чергу, SQLite не має вбудованих серверних інструментів для резервного копіювання або реплікації. Копіювання файлу бази даних здійснюється на рівні файлової системи, що не гарантує транзакційної цілісності у момент запису. Реплікація в класичному розумінні не підтримується, тому SQLite доцільно використовувати лише у локальних або не критичних до відмовостійкості застосунках [25].

Підтримка JSON- та NoSQL-функціональності. Підтримка роботи з напівструктурованими даними (JSON) і елементів NoSQL-функціональності теж є важливим критерієм для сучасних інформаційних систем, які часто потребують зберігання даних у вигляді гнучких та динамічних структур.

PostgreSQL є беззаперечним лідером серед реляційних СУБД за рівнем підтримки JSON. Система має повноцінний тип даних `JSONB`, дозволяє зберігати, індексувати та виконувати складні операції над JSON-документами, а також здійснювати складні запити, агрегацію, фільтрацію та використовувати оператори. Завдяки цьому PostgreSQL активно використовується як у класичних реляційних, так і в гібридних рішеннях [13, 20].

СУБД MySQL з версії 5.7 також підтримує тип даних JSON, що дає змогу зберігати документи у таблиці та виконувати базові операції над ними (пошук, оновлення і валідацію). Водночас індексація та аналітичні функції для JSON реалізовані не так гнучко, як у PostgreSQL, тому система часто використовується

для реалізації часткової NoSQL-архітектури, коли для виконання поставлених завдань достатньо простих методів роботи з напівструктурованими даними [12, 18].

СУБД SQLite підтримує роботу з JSON на базовому рівні завдяки розширенню JSON1. В ньому реалізовані функції для парсингу, пошуку та зміни елементів JSON-структур. Однак спеціалізованої індексації чи підтримки складних запитів не передбачено, а отже, повноцінної NoSQL-функціональності ця СУБД не забезпечує [25].

Firebird демонструє помітне вдосконалення у цьому аспекті з виходом версії 5.0, у якій реалізовано базу підтримку типу даних JSON та простих операцій над ним. Можливості поки що обмежені порівняно з PostgreSQL і MySQL, проте для більшості корпоративних і освітніх застосунків цього достатньо. Через те, для проєктів, які не потребують глибокої NoSQL-аналітики, це не є критичним обмеженням [14].

Підтримка транзакцій (ACID) та багатoversійність. Підтримка транзакцій та дотримання принципів ACID (Atomicity, Consistency, Isolation, Durability) є надважливою вимогою для корпоративних і освітніх систем, де критично важливе забезпечення цілісності даних.

СУБД Firebird відома саме якісною реалізацією транзакційної моделі: система забезпечує повну підтримку ACID і має потужний механізм багатoversійного управління конкурентністю (MVCC). Такий механізм дозволяє ефективно обробляти складні бізнес-процеси та гарантувати цілісність навіть при великій кількості паралельних транзакцій, оскільки знижується ризик блокувань ресурсів бази даних. Саме завдяки цьому Firebird часто обирають для рішень із підвищеними вимогами до захищеності та надійності даних [14, 15, 22].

СУБД PostgreSQL також є зразковим прикладом реалізації ACID і MVCC, оскільки тут підтримуються всі рівні ізоляції, ефективно здійснюється обробка конкурентних запитів та існують механізми захисту від «фантомних» читань. Окрім цього, PostgreSQL пропонує гнучкі можливості як для вертикального, так і для горизонтального масштабування, засоби кластеризації та побудови складних розподілених рішень [13, 20].

MySQL до появи ядра InnoDB поступалася лідерам нашого аналізу у цьому питанні, однак із InnoDB СУБД повністю забезпечує підтримку транзакцій та принципів ACID. Система також добре масштабується для вебзастосунків, де важливими критеріями є швидкість і можливість горизонтального розширення. Проте для складних транзакційних і аналітичних сценаріїв PostgreSQL і Firebird часто демонструють кращу цілісність та контроль ізоляції [12, 18].

SQLite реалізує підтримку базових транзакцій у відповідності до принципів ACID, зокрема у режимі WAL-журналювання. Однак ця СУБД розрахована на прості автономні сценарії, має обмеження щодо паралельних записів та не підтримує складне масштабування чи багатокористувацьку роботу з високим рівнем конкурентності [25].

Таким чином, Firebird та PostgreSQL оптимальні для систем з високими вимогами до транзакційної цілісності, складної бізнес-логікою та високими вимогами до надійності, MySQL – для масштабованих вебрішень, а SQLite – для вбудованих і автономних застосунків із невеликою кількістю одночасних клієнтів.

Інструменти розробки, повнотекстовий пошук і геопросторові можливості. Рівень розвитку інструментів для розробки, а також наявність спеціалізованого функціоналу (наприклад, повнотекстовий пошук чи підтримка геопросторових даних) відображають зрілість екосистеми кожної СУБД.

Екосистема PostgreSQL наразі пропонує найширший спектр інструментів: для адміністрування використовуються pgAdmin, DBeaver, Navicat, CLI та велика кількість сторонніх IDE-середовищ. Для повнотекстового пошуку система має потужну вбудовану реалізацію із підтримкою багатомовності, індексації та ранжування. Щодо геопросторових можливостей, то PostgreSQL з модулем PostGIS де факто став «золотим стандартом» для GIS-проєктів і наукових досліджень, оскільки в такому комплекті стає доступним майже повний набір стандартів OpenGIS [13, 20].

СУБД MySQL має власні зручні інструменти адміністрування (Workbench, phpMyAdmin, DBeaver), достатньо розвинене підтримку повнотекстового пошуку (з ядрами MyISAM та InnoDB) та базові можливості роботи з геопросторовими даними. Водночас ці можливості поступаються за гнучкістю та масштабованістю рішенням на базі PostgreSQL [12, 18].

СУБД SQLite орієнтована на простоту, а тому для адміністрування і розробки достатньо консольного інтерфейсу або легких інструментів типу DB Browser for SQLite чи DBeaver). Для повнотекстового пошуку доступні розширення FTS3/FTS4/FTS5, а геопросторовий функціонал реалізується через SpatiaLite. Однак все це залишається в рамках автономних, або малонавантажених систем [25].

Firebird сфокусований на якісному базовому функціоналі. Адміністрування підтримується через FlameRobin, IBExpert, DBeaver, зручний CLI, широкий набір засобів моніторингу і резервування. Повнотекстовий пошук в СУБД реалізується лише за допомогою сторонніх рішень або розширень, а геопросторові функції відсутні у базовій комплектації, через те для таких завдань застосовують інтеграцію з окремими сервісами або спеціалізованим програмним забезпеченням [14, 15].

Загальні результати проведеного дослідження. Аналіз відкритих СУБД за основними критеріями продемонстрував, що жодна з досліджуваних систем не є універсальним рішенням для побудови

інформаційної системи ЗВО. Кожна з розглянутих СУБД має власні сильні сторони, недоліки, обмеження та сферу застосування. За результатами аналізу сформовано підсумкову таблицю для оцінювання привабливості досліджуваних відкритих СУБД для ЗВО (таблиця 2).

Таблиця 2.

Загальна оцінка функціональних можливостей відкритих СУБД

СУБД	Основні переваги	Основні обмеження	Сумарна кількість балів
PostgreSQL	Розширена підтримка SQL-стандартів, багатofункціональність, масштабованість, повна ACID-сумісність, розвинена робота з JSON/NoSQL, потужна спільнота, гнучкість для аналітики та наукових проєктів, підтримка хмарних сервісів.	Потребує досвідчених адміністраторів, значні апаратні ресурси при великих навантаженнях, горизонтальне масштабування часто вимагає зовнішніх рішень.	67
MySQL	Висока продуктивність для вебсистем, простота розгортання, велика спільнота, розвинена реплікація, підтримка популярних інструментів	Менша відповідність SQL-стандартам, менш потужний аналітичний функціонал, питання ліцензування для комерційних проєктів, потребує оптимізації при масових INSERT	59
Firebird	Відкритий код, простота розгортання, кросплатформність, якісна транзакційна модель, ефективність на середніх і малих ресурсах, просте адміністрування	Обмежена NoSQL-функціональність (базова підтримка JSON), відсутність вбудованої кластерної реплікації, менша екосистема, спеціалізована спільнота	47
SQLite	Компактність, нульова конфігурація, портативність, ACID, ефективна для мобільних/вбудованих систем	Обмежена паралельність запису, відсутність інструментів масштабування/реплікації, немає власної аутентифікації	37

Таким чином, на нашу думку, відповідно до підсумкової таблиці 2 можна стверджувати наступне:

1. СУБД PostgreSQL впевнено лідирує за функціональністю, відповідністю SQL-стандартам, можливостями до розширення, підтримкою транзакцій та багатoversійності, а також функціоналом для аналітики, повнотекстового пошуку та роботи з напівструктурованими даними (JSON/NoSQL). При цьому система потребує висококваліфікованого адміністрування та значних апаратних ресурсів, а горизонтальне масштабування вимагає залучення сторонніх рішень.

2. MySQL вирізняється простотою використання, швидкою обробкою транзакцій у стандартних вебсценаріях, широкими засобами реплікації, активною підтримкою спільноти та високою доступністю інструментів розробки та адміністрування. Основними обмеженнями є менша відповідність SQL-стандартам, складніша організація паралельних операцій запису та потенційні ліцензійні особливості у комерційних проєктах.

3. СУБД SQLite є кращим вибором для автономних, портативних та мобільних застосунків, де надзвичайно важливими є компактність, відсутність потреби в адмініструванні і простота використання. Серед обмежень можна виділити низький рівень управління конкурентними операціями запису, відсутність засобів управління доступом та інструментів масштабування і реплікації.

4. Firebird поєднує корпоративний рівень надійності з простотою розгортання, мінімальними витратами на адміністрування та відкритістю коду. СУБД вирізняється якісною транзакційною моделлю, кросплатформністю, компактністю та ефективністю роботи у середовищах із обмеженими ресурсами. Водночас варто враховувати дещо обмежену підтримку сучасних NoSQL-функцій, відсутність вбудованої реплікації кластерного рівня та порівняно невелику спільноту розробників.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

Виходячи з проведеного аналізу, на нашу думку, можна сформулювати наступні практичні рекомендації для вибору відкритої СУБД при створенні чи модернізації інформаційної системи ЗВО:

1. Проведення ретельного аналізу запланованого робочого навантаження. Вибір СУБД має ґрунтуватися в залежності від типів домінуючих операцій (читання, запис, оновлення, видалення), прогнозованих обсягів даних для обробки, а також рівня конкуруючих операцій та складності запитів.

2. Оцінювання сукупної вартості володіння СУБД. Оскільки мова йде про відкриті системи, то потрібно розуміти, що відсутність плати за придбання ліцензії зовсім не нівелює витрати на розгортання, обслуговування, оптимізацію, потребу в залученні кваліфікованого персоналу та можливі потенційні витрати на різноманітні комерційні розширення та інструменти.

3. Планування майбутніх вимог до масштабованості. Якщо передбачається значне зростання обсягів даних або кількості користувачів, то необхідно спробувати оцінити, наскільки легко СУБД може масштабуватися горизонтально або вертикально і які інструменти чи архітектурні зміни для цього потрібні.

4. Вивчення екосистеми та спільноти продукту. Порівняно велика та активна спільнота, достатні обсяги і рівні деталізації документованих функцій, а також широкий спектр кількості сторонніх інструментів можуть значно спростити розробку, розгортання і усунення несправностей у системі.

5. Орієнтування на стабільність та безпеку. Для критично важливих систем стабільність та надійні засоби захисту даних є першочерговими. Також при переході на новіші версії важливо здійснювати ретельне тестування рішень перед їхнім застосуванням у виробничому середовищі.

Таким чином, успішний вибір відкритої СУБД для інформаційної системи ЗВО передбачає системний підхід, зважене порівняння сукупних переваг, недоліків та обмежень кожної системи, аналіз специфіки поставлених завдань і наявних ресурсів закладу освіти.

До перспективних напрямів подальших досліджень у сфері використання відкритих СУБД в інформаційних системах ЗВО ми вбачаємо у поглибленому аналізі їх продуктивності та масштабованості з врахуванням реальних сценаріїв навантаження, інтеграції з хмарними платформами, використання контейнерних технологій, а також вивчення досвіду міграції з пропрієтарних рішень до відкритих систем із мінімізацією витрат і ризиків для освітніх установ.

Література

1. Глазунова О. М. Веб-орієнтована система електронного деканату: реалізація прецеденту формування індивідуального плану студента / О.М. Глазунова, В. М. Андрущенко, Т. В. Волошина, В.І. Корольчук // Інформаційні технології і засоби навчання. – 2024, №1(99). – С. 230–244. DOI: 10.32689/maur.it.2024.2.4

2. Косіюк М.М. Автоматизована інформаційна система управління закладом вищої освіти «Електронний університет» / М. М. Косіюк, К.Е. Більовський, В.М. Лисак // Інформаційні технології і засоби навчання. – 2023, Т. 93, №1. – С. 96–116. DOI: 10.33407/itlt.v93i1.5107

3. Кравчина О.Є. Основні напрями використання вільного програмного забезпечення в закладах освіти зарубіжжя / О. Є. Кравчина // Інформаційні технології і засоби навчання. – 2010. – №6(20). – С. 12–19. doi: 10.33407/itlt.v20i6.372

4. Мокрієв М.В. Інтеграція навчально-наукових підсистем в єдине інформаційно-освітнє середовище (на базі відкритого програмного забезпечення) / М.В. Мокрієв // Відкрите освітнє е-середовище сучасного університету. – 2020. – №8. – С. 60–71. DOI: 10.28925/2414-0325.2020.8.7

5. Триус Ю.В. Інформаційно-аналітична система підтримки освітньої діяльності структурних підрозділів закладів вищої освіти / Ю.В. Триус, Г. О. Заска, О. С. Кожем'якін, А. В. Аширова // Вісник Черкаського державного технологічного університету. – 2020. – №4. – С. 27–38. DOI: 10.24025/2306-4412.4.2020.219482.

6. Ancuta N. Compare Firebird SQL vs Mysql / N. Ancuta, M.Hussain, N. Tiwari – Available at: <https://www.peerspot.com/products/comparisons/firebird-sql-vs-mysql-vs-sqlite>

7. Beach P. Migration to Firebird 3: Step by Step / P. Beach // Firebird Conference Proceedings. – Prague, 2017. – P. 45–67.

8. Borrie H. The Firebird Book: A Reference for Database Developers / H. Borrie ; 2nd ed. – Berkeley: Apress, 2004. – 1104 p.

9. Choi N. The context and state of open source software adoption in US academic libraries N. Choi, J.A. Pruetz // Library Hi Tech. – 2019, V. 37, No. 4. – P. 641–659. <https://doi.org/10.1108/LHT-02-2019-0042>

10. DB-Engines Ranking Open Source vs. Commercial DBMS. – Available at: https://db-engines.com/en/ranking_osvsc

11. Drake M. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. – Available at: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>

12. Essential MySQL. – URL: <https://www.programming-books.io/essential/mysql/>

13. Essential PostgreSQL. – Available at: <https://www.programming-books.io/essential/postgresql/>

14. Firebird: Firebird RDBMS / Firebird Project. – Available at: <https://firebirdsql.org/en/documentation/>

15. FirebirdFAQ.org – up-to-date answers to Firebird related questions. – Available at: <https://www.firebirdfaq.org/>

16. Kowalczyk R. Performance analysis of selected database management systems / R. Kowalczyk, J. Smółka // Journal of Computer Sciences Institute. – 2024. – № 32. – P. 151–156. <https://doi.org/10.35784/jcsi.6224>

17. Manzur W. Getting Started with SQLite. Boston University / W. Manzur. – 2021. – 41 p. – Available at: <https://www.bu.edu/csmet/files/2021/03/Getting-Started-with-SQLite.pdf>

18. MySQL documentation. – Available at: <https://dev.mysql.com/doc/>

19. MySQL Notes for Professionals. – Available at: <https://goalkicker.com/MySQLBook>

20. PostgreSQL: Documentation. – Available at: <https://www.postgresql.org/docs/>

21. PostgreSQL Notes for Professionals. – Available at: <https://goalkicker.com/PostgreSQLBook>

22. Rodriguez M. Transaction Management in Firebird: Best Practices / M. Rodriguez // Database Systems Journal. – 2022. – Vol. 13, No. 2. – P. 34–51.
23. Salunke S. V. A Performance Benchmark for the PostgreSQL and MySQL Databases / S.V. Salunke, A. Ouda // Future Internet. – 2024, 16(10). <https://doi.org/10.3390/fi16100382>
24. Silén R. Why Switch.ch Chose MariaDB: A Swiss Case Study in Open Source, Digital Sovereignty, and Academic IT. – Available at: <https://mariadb.org/swiss-academic-it-switch-ch/>
25. SQLite documentation. – Available at: <https://www.sqlite.org/docs.html>
26. SQLite vs MySQL VS PostgreSQL (detailed comparison). – Available at: <https://runcloud.io/blog/sqlite-vs-mysql-vs-postgresql>
27. Yohannes W. The Practice of Opting for Open Source Solutions in Higher Education Institutions of Ethiopia. – 2012. – 108 p. – Available at: <https://etd.aau.edu.et/server/api/core/bitstreams/a7717fa0-a504-4bda-aea8-a1f437d2af3d/content>
28. Zapata J.G. MySQL vs PostgreSQL: A Comparative Analysis of RDBMS Technologies Response Time in Web-based E-commerce / J.G. Zapata // ResearchGate Preprint. – 2024. <https://doi.org/10.13140/RG.2.2.24791.69288>

References

1. Hlazunova O.M. Web-oriented system of the electronic dean's office: implementation of the precedent of forming the student's individual plan / O.M. Hlazunova, V.M. Andiuschenko, T.V. Voloshyna, V.I. Koroliuk // Information Technology and Society. – 2024, №1(99). – C. 230–244. DOI: <https://doi.org/10.32689/maup.it.2024.2.4>
2. Kosiyuk M.M. Automated information management system of higher education institution «Electronic University» / M. M. Kosiyuk, K. E. Bilovskiy, V. M. Lysak // Information Technologies and Learning Tools. – 2023, T. 93, №1. – C. 96–116. DOI: <https://doi.org/10.33407/itlt.v93i1.5107>
3. Kravchyna O.Y. Main approaches of free software application in educational institutions abroad / O. Y. Kravchyna // Information Technologies and Learning Tools. – 2010. – №6(20). – C. 12–19. <https://doi.org/10.33407/itlt.v20i6.372>
4. Mokriiev M. V. Integration of educational subsystems in all-in-one information and education environment / M. V. Mokriiev // Open educational e-Environment of modern University. – 2020. – №8. – C. 60–71. <https://doi.org/10.28925/2414-0325.2020.8.7>
5. Tryus Yu. V. Information and analytical system for educational activities support of structural divisions of higher education institutions / Yu. V. Tryus, H. O. Zaspá, O.S. Kozhemiakin, A. V. Ashirova // Bulletin of Cherkasy State Technological University. – 2020. – №4. – C. 27–38. DOI: <https://doi.org/10.24025/2306-4412.4.2020.219482>
6. Ancuta N. Compare Firebird SQL vs Mysql / N. Ancuta, M.Hussain, N. Tiwari – Available at: <https://www.peerspot.com/products/comparisons/firebird-sql-vs-mysql-vs-sqlite>
7. Beach P. Migration to Firebird 3: Step by Step / P. Beach // Firebird Conference Proceedings. – Prague, 2017. – P. 45–67.
8. Borrie H. The Firebird Book: A Reference for Database Developers / H. Borrie ; 2nd ed. – Berkeley: Apress, 2004. – 1104 p.
9. Choi N. The context and state of open source software adoption in US academic libraries N. Choi, J.A. Pruett // Library Hi Tech. – 2019, V. 37, No. 4. – P. 641–659. <https://doi.org/10.1108/LHT-02-2019-0042>
10. DB-Engines Ranking Open Source vs. Commercial DBMS. – Available at: https://db-engines.com/en/ranking_osvsc
11. Drake M. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. – Available at: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
12. Essential MySQL. – URL: <https://www.programming-books.io/essential/mysql/>
13. Essential PostgreSQL. – Available at: <https://www.programming-books.io/essential/postgresql/>
14. Firebird: Firebird RDBMS / Firebird Project. – Available at: <https://firebirdsql.org/en/documentation/>
15. FirebirdFAQ.org – up-to-date answers to Firebird related questions. – Available at: <https://www.firebirdfaq.org/>
16. Kowalczyk R. Performance analysis of selected database management systems / R. Kowalczyk, J. Smolka // Journal of Computer Sciences Institute. – 2024. – № 32. – P. 151–156. <https://doi.org/10.35784/jcsi.6224>
17. Manzur W. Getting Started with SQLite. Boston University / W. Manzur. – 2021. – 41 p. – Available at: <https://www.bu.edu/csmet/files/2021/03/Getting-Started-with-SQLite.pdf>
18. MySQL documentation. – Available at: <https://dev.mysql.com/doc/>
19. MySQL Notes for Professionals. – Available at: <https://goalkicker.com/MySQLBook>
20. PostgreSQL: Documentation. – Available at: <https://www.postgresql.org/docs/>
21. PostgreSQL Notes for Professionals. – Available at: <https://goalkicker.com/PostgreSQLBook>
22. Rodriguez M. Transaction Management in Firebird: Best Practices / M. Rodriguez // Database Systems Journal. – 2022. – Vol. 13, No. 2. – P. 34–51.
23. Salunke S. V. A Performance Benchmark for the PostgreSQL and MySQL Databases / S.V. Salunke, A. Ouda // Future Internet. – 2024, 16(10). <https://doi.org/10.3390/fi16100382>
24. Silén R. Why Switch.ch Chose MariaDB: A Swiss Case Study in Open Source, Digital Sovereignty, and Academic IT. – Available at: <https://mariadb.org/swiss-academic-it-switch-ch/>
25. SQLite documentation. – Available at: <https://www.sqlite.org/docs.html>
26. SQLite vs MySQL VS PostgreSQL (detailed comparison). – Available at: <https://runcloud.io/blog/sqlite-vs-mysql-vs-postgresql>
27. Yohannes W. The Practice of Opting for Open Source Solutions in Higher Education Institutions of Ethiopia. – 2012. – 108 p. – Available at: <https://etd.aau.edu.et/server/api/core/bitstreams/a7717fa0-a504-4bda-aea8-a1f437d2af3d/content>
28. Zapata J.G. MySQL vs PostgreSQL: A Comparative Analysis of RDBMS Technologies Response Time in Web-based E-commerce / J.G. Zapata // ResearchGate Preprint. – 2024. <https://doi.org/10.13140/RG.2.2.24791.69288>