

БУБНОВСЬКА ІРИНА

Вінницький національний аграрний університет

<https://orcid.org/0000-0002-3535-3127>e-mail: gira25100@gmail.com

ЗАСТОСУВАННЯ МЕТОДІВ ЛІНІЙНОЇ АЛГЕБРИ ДЛЯ ОПТИМІЗАЦІЇ ЗАПИТІВ У СИСТЕМАХ УПРАВЛІННЯ БАЗАМИ ДАНИХ

У статті досліджено застосування методів лінійної алгебри для оптимізації запитів у сучасних системах управління базами даних. Запропонований підхід полягає у формалізації табличних структур даних як числових матриць, що дозволяє інтерпретувати реляційні операції як лінійно-алгебраїчні перетворення, що дозволяє застосовувати математичні інструменти для оптимізації обчислень. Розглянуто основні методи лінійної алгебри для оптимізації запитів – матричну факторизацію (LU, QR-розклад) для структурування складних обчислень; сингулярний розклад матриці (SVD) для зменшення розмірності і виявлення прихованих залежностей; матричне моделювання JOIN-операцій для ефективного блокування; системи лінійних рівнянь для формалізації фільтрації даних. Для кожного методу описано принцип дії, переваги та обмеження, а також типові сфери застосування. Проведено експериментальне дослідження на синтетичному наборі даних, що імітує транзакційну таблицю з мільйоном записів. Зроблено висновок, що лінійна алгебра є потужною формалізаційною основою для архітектур СУБД нового покоління, здатною забезпечити математичну узгодженість, обчислювальну ефективність і масштабованість.

Ключові слова: лінійна алгебра, оптимізація запитів, системи управління базами даних, матричні обчислення, SVD, JOIN-операції.

BUBNOVSKA IRYNA

Vinnytsia National Agrarian University

APPLICATION OF LINEAR ALGEBRA METHODS TO OPTIMISE QUERIES IN DATABASE MANAGEMENT SYSTEMS

The article investigates the use of linear algebra methods for query optimisation in modern database management systems. The relevance of the study is due to the growth of data volumes and the need for more efficient methods of their processing. The proposed approach is to formalise tabular data structures as numerical matrices, which allows us to interpret relational operations (selection, projection, join) as linear algebraic transformations, which opens up opportunities for the use of powerful mathematical tools that optimise computations. The main methods of linear algebra for query optimisation are considered: matrix factorisation (LU, QR decomposition) for structuring complex calculations; singular value decomposition (SVD) for reducing dimensionality and identifying hidden dependencies; matrix modelling of JOIN operations as an efficient block multiplication; systems of linear equations for formalising data filtering. It is noted that matrix factorisation helps to optimise the execution of complex analytical queries, SVD is effective in aggregation and dimensionality reduction tasks, the matrix representation of JOIN provides high performance when processing large amounts of data, and linear systems unify the filtering logic with numerical processing. To empirically verify the proposed methods, an experimental study was conducted on a synthetic dataset simulating a transactional table with a million records. The experiment was carried out using the NumPy and PyTorch libraries with the use of GPU acceleration. We evaluated such indicators as query execution time, RAM usage, and scalability. The results showed that JOIN matrix modelling was the fastest, which is especially important for parallel computing. LU/QR factorisation demonstrated stability and versatility, while SVD proved to be resource-intensive but valuable for data compression. Linear equation methods showed balanced performance. It is concluded that linear algebra is a powerful formalisation framework for next-generation DBMS architectures capable of providing mathematical consistency, computational efficiency and scalability.

Keywords: linear algebra, query optimisation, database management systems, matrix computing, SVD, JOIN operations.

Стаття надійшла до редакції / Received 01.09.2025

Прийнята до друку / Accepted 15.11.2025

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

У сучасних інформаційних системах обсяг, складність і швидкість обробки даних стрімко зростають, що зумовлює потребу в удосконаленні методів оптимізації запитів у системах управління базами даних (СУБД). Традиційні підходи до оптимізації, засновані на правилах реляційної алгебри або евристичних механізмах планування виконання запитів, часто виявляються недостатньо ефективними в умовах великих обсягів даних, високої частоти звернень або розподіленого середовища. Актуальності набувають методи лінійної алгебри, які дозволяють формалізувати структури та операції над даними у вигляді матриць, векторів і систем рівнянь, що відкриває можливості для векторизації обчислень, паралельної обробки та зменшення обчислювальної складності. Прогрес у галузі обчислювальної алгебри, GPU-прискорення та інтеграція машинного навчання в СУБД сприяють активному впровадженню лінійно-алгебраїчних моделей у процес обробки запитів. Проте, не зважаючи на потенційні переваги, існує обмежена кількість досліджень, що системно описують алгоритмічні та архітектурні аспекти впровадження таких методів у прикладні середовища СУБД.

Проблема полягає в недостатньо розробленій методології застосування методів лінійної алгебри для оптимізації запитів у СУБД. Зокрема, бракує уніфікованих моделей представлення табличних структур у вигляді матриць, формалізованих процедур перетворення SQL-запитів у алгебраїчні операції та алгоритмів ефективного виконання цих операцій у реальних умовах. Виникає потреба у визначенні переваг, обмежень і практичних сценаріїв застосування таких підходів, а також у розробленні експериментальних моделей, що

дозволяють оцінити їх продуктивність у порівнянні з класичними методами оптимізації запитів.

Аналіз досліджень та публікацій

У сучасних дослідженнях спрямованих на підвищення ефективності запитів у системах управління базами даних, зростає значення підходів заснованих на використанні методів лінійної алгебри. Особливої уваги заслуговують дослідження, у яких лінійна алгебра використовується як інструмент представлення табличних структур і перетворення запитів. Так, алгебраїчна формалізація й факторизаційні підходи розглянуто у роботі Суліми С.В., Єрмолаєва О.Д. [1], де запропоновано метод оптимізації SQL-запитів, що включає розгляд структур даних у вигляді матриць і дозволяє формалізувати операції над запитами як лінійні перетворення, які є основою для подальшої обчислювальної оптимізації. У дослідженні Dudar Z., Liashyk V. [2] обґрунтовано можливість використання алгебри лінійних предикатів для розв'язання систем кванторних рівнянь, що забезпечує математичну формалізацію умов у запитах типу WHERE та фільтрації. Ефективність виконання запитів, пов'язаних із машинним навчанням, через перетворення SQL-операцій у лінійно-алгебраїчні процедури, що дозволяє реалізувати запити за допомогою GPU-архітектур продемонстровано Sun W., Katsifodimos A., Hai R. [3].

Алгебру запитів для нестандартних форматів розглянуто Langerak A.J., Frasincar F., Klinkhamer J. [4], які запропонували JAL-алгебру для оптимізації JSON-запитів. Зінченко П., Мехед Д. [5] вивчили оптимізацію запитів у GraphQL, приділивши увагу аналітичним показникам продуктивності при запитуванні графових структур – підхід, який також можливо формалізувати в термінах лінійної алгебри.

Розширення арсеналу методів оптимізації від строгих алгебраїчних моделей до інтелектуальних евристик, що створює передумови для підвищення ефективності запитів у великих та розподілених інформаційних системах здійснено у працях Fent Ph., Moerkotte G., Neumann Th. [6], Gaikwad T., Mohite A., Tarawade M. [7], Veerappa P. M., Chikkamannur A. A. [8] та ін.

Дослідження Chatterjee N., Sharma M., Sood S., Shubneet, Yadav A.R. [9], Guo Y., Li G., Hu R., Wang Y. [10], Fahima M.M.F., Sahna Sreen A.H., Fathima Ruksana S.L., Weihsen D.T.E., Majid M.H.M. [11], Ma X. [12] та ін. демонструють еволюцію підходів до оптимізації запитів від традиційних реляційних моделей до інтелектуально керованих СУБД, де поєднуються методи машинного навчання, глибоких нейронних мереж та предиктивної аналітики.

Проблеми підвищення продуктивності СУБД шляхом аналізу конфігураційних параметрів, кешування та профілювання запитів досліджуються Семенюк В.В. [13], Ільїна І.В., Кадубенко Д.С., Ільїн Д.О. [14] та ін. Белоус Р.В., Крилов Є.В. [15] запропонували вдосконалення використання алгоритму Raft у розподілених СУБД, що безпосередньо впливає на стабільність та ефективність виконання запитів у кластерах.

Аналіз наукових праць свідчить, що методи лінійної алгебри послідовно інтегруються у сучасні підходи до оптимізації роботи СУБД. Спостерігається еволюція рішень від класичних факторизаційних методів до інтелектуальних технологій. Поєднання засобів, що охоплюють сингулярний розклад, комбінаторні алгоритми та системи штучного інтелекту, демонструє міждисциплінарний характер проблематики та підтверджує потенціал лінійної алгебри як основи для підвищення ефективності обробки даних у великих та розподілених інформаційних системах.

Формулювання цілей статті

Метою роботи є обґрунтування підходів до оптимізації запитів у системах управління базами даних на основі методів лінійної алгебри. Дослідження спрямоване на формалізацію табличних структур у вигляді матричних моделей, побудову та апробацію алгоритмів виконання базових операцій реляційної алгебри як лінійно-алгебраїчних трансформацій, а також на експериментальне оцінювання ефективності таких методів за показниками часу виконання, споживання пам'яті та масштабованості.

Виклад основного матеріалу

Сучасні системи управління базами даних дедалі частіше інтегрують методи прикладної математики, зокрема лінійної алгебри, для формалізації та оптимізації запитів. Матричне представлення структур даних дає змогу не лише ефективно відтворювати операції реляційної алгебри, а й виконувати їх за допомогою високопродуктивних лінійно-алгебраїчних обчислень, особливо у середовищах із паралельною обробкою та GPU-прискоренням.

Важливим етапом упровадження лінійної алгебри в оптимізацію запитів є формалізація табличних структур у вигляді матриць, що створює підґрунтя для перетворення реляційних операцій у лінійно-алгебраїчні процедури, істотно підвищуючи ефективність їх виконання. З метою реалізації такого підходу розглянемо принципи побудови матричних моделей для представлення табличних структур. Сформулюємо спосіб відображення реляційної таблиці у вигляді числової матриці, що стане основою для подальшого застосування лінійно-алгебраїчних операцій: кожен запис у базі даних можна подати як числову матрицю $A \in R_{n \times m}$, де n – кількість записів (рядків); m – кількість атрибутів (стовпців).

Таке представлення дозволяє реалізовувати реляційні операції, такі як селекція, проєкція, об'єднання та агрегація, у вигляді матричних трансформацій або систем лінійних рівнянь.

На основі узагальненого матричного подання таблиць стає можливим застосування різноманітних методів лінійної алгебри до моделювання, трансформації та оптимізації запитів. Розглянемо детальніше найбільш поширені з них, починаючи з методів матричної факторизації, що дають змогу структуровано організовувати обчислення:

1. Метод LU/QR-факторизації доцільно використовувати у задачах попередньої обробки даних або

аналітики. У сучасних СУБД цей метод частіше застосовується у вбудованих ML-модулях або при виконанні складних обчислювальних запитів у системах з аналітичним або OLAP-фокусом.

2. Метод використання сингулярного розкладу матриці (SVD) дозволяє зменшити розмірність запитів шляхом виявлення латентної структури у таблиці. Його результати можуть бути використані в задачах тематичного групування або попередньої обробки, що знижує обсяг повторних обчислень і підвищує ефективність при побудові матеріалізованих подань. SVD актуальний для оптимізації запитів у великих таблицях, що містять надлишкові атрибути або повторювані записи. SVD також застосовується у побудові матеріалізованих подань або кешування результатів агрегатів, що часто використовуються.

3. Метод матричного моделювання JOIN-операції за певних умов (наявність числових і закодованих ключів) можна апроксимувати через матричний добуток з умовним фільтром:

$$R \bowtie S \approx (R \times S^T) \circ M, \quad (1)$$

де M – маска умов приєднання.

Такий підхід ефективний у GPU-середовищі або для задач із векторизованою обробкою.

4. Метод систем лінійних рівнянь для фільтрації використовується для числових або лінійно виражених умов. Запити типу WHERE можна інтерпретувати як системи лінійних нерівностей, які задають гіперплощини у векторному просторі, що дозволяє реалізовувати фільтрацію через маскування або логічні операції над матрицями. Для складних умов доцільне попереднє кодування або комбінований підхід.

Наведені вище методи демонструють різноманіття підходів до оптимізації запитів у системах управління базами даних із використанням лінійної алгебри. Кожен із них має власну специфіку застосування, переваги та обмеження, які залежать від типу даних, складності запиту та архітектури обчислювального середовища. У табл. 1 проаналізовано основні характеристики, що дозволить більш обґрунтовано обирати відповідний метод у практичних сценаріях реалізації.

Таблиця 1

Порівняльна характеристика методів лінійної алгебри для оптимізації запитів у СУБД

Методи лінійної алгебри	Основні переваги	Основні обмеження	Типові сфери застосування
LU , QR -факторизація	Структурування обчислень, зменшення складності вкладених підзапитів	Висока складність реалізації, потреба у числовому представленні даних	Аналітичні запити, підзапити, багаторівневі об'єднання
Сингулярний розклад матриці (SVD)	Зменшення розмірності, виявлення латентної структури, семантичне узагальнення	Високі обчислювальні витрати, складність реалізації у реальному часі	Побудова агрегатів, кешування, тематичний аналіз запитів
Матричне моделювання операцій JOIN	Висока продуктивність, можливість векторизації та паралелізації об'єднань	Висока потреба в оперативній пам'яті, чутливість до розміру таблиць	Масштабні об'єднання таблиць у розподілених СУБД
Системи лінійних рівнянь для фільтрації	Формалізація логічних умов, уніфікація фільтрації та числової обробки	Обмежена виразність для складних умов, залежність від попереднього кодування	Селекція за умовами WHERE, семантична фільтрація

Аналіз методів свідчить про багатоваріантність лінійно-алгебраїчних підходів до оптимізації запитів у системах управління базами даних. Кожен метод характеризується власним балансом між обчислювальною ефективністю, вимогами до ресурсів та специфікою застосування.

Отже, лінійна алгебра постає як потужна формалізаційна основа, здатна забезпечити математичну узгодженість, обчислювальну ефективність і масштабованість при оптимізації запитів у сучасних інформаційних системах.

Для практичної верифікації можливостей застосування лінійної алгебри до оптимізації запитів у СУБД розглянемо експериментальний прототип системи, що моделює базу даних із векторизованим обчислювальним ядром на основі методів лінійної алгебри, подібно до архітектур, які застосовуються у GPU-прискорених СУБД або tensor-oriented системах. У цій системі всі табличні структури та операції реалізуються у вигляді матричних подань та обчислень, що дозволяє зіставити продуктивність традиційного підходу і лінійно-алгебраїчного підходу до виконання запитів.

Зокрема, експериментальний прототип охоплює такі типи запитів:

- селекцію записів за умовами (WHERE) через побудову масок та їх застосування до матриць;
- JOIN-операції на основі блочних матричних добутоків із фільтрацією за умовною маскою;
- агрегацію та проєкцію, як матричні операції на підмножинах стовпців;
- зменшення розмірності (SVD) для виявлення тематичної структури запитів.

Для забезпечення об'єктивного порівняння ефективності розглянутих лінійно-алгебраїчних підходів було проведено серію обчислювальних експериментів із фіксованим обсягом даних, єдиними умовами виконання та уніфікованим апаратним середовищем. Експеримент проведено на синтетичному наборі даних, що імітує типову транзакційну таблицю з 1 млн записів та 20 атрибутів. Для реалізації матричних операцій використовувались GPU NVIDIA RTX 3060 (12 GB VRAM), CPU Intel Core i7, 32 GB RAM. Для матричних обчислень використовувались бібліотеки NumPy 1.26 і PyTorch 2.1, GPU-прискорення активовано через CUDA 12.1. Кожен експеримент проводився тричі, а в табл. 2 наведено середні значення. Всі запити виконувались у

контрольованому середовищі із фіксованими апаратними параметрами.

Отримані результати для кожного методу охоплюють основні характеристики обробки запитів – час виконання, обсяг споживаної оперативної пам'яті та здатність методів до масштабування при зростанні розміру вхідних таблиць, що дає змогу порівняти їх ефективність відповідно до зазначених критеріїв і визначити оптимальні сценарії використання кожного із підходів у практичній реалізації СУБД (табл. 2).

Таблиця 2

Експериментальні результати продуктивності лінійно-алгебраїчних трансформацій запитів у СУБД

Метод	Час виконання (сек)	Пам'ять (МБ)	Масштабованість (0–10)*
<i>LU/QR</i> -факторизація	1,8	520	7,5
Сингулярний розклад (<i>SVD</i>)	2,5	980	6,8
Матричне моделювання <i>JOIN</i>	1,2	750	8,2
Системи лінійних рівнянь (<i>WHERE</i>)	1,6	610	7,0

*Показник масштабованості визначено емпірично на основі приросту часу виконання при збільшенні обсягу даних у 10 разів. Значення 10 відповідає ідеальній масштабованості, коли приріст часу не перевищує лінійного.

Аналіз отриманих результатів показав, що найвищу ефективність за показником часу виконання продемонструвало матричне моделювання *JOIN*-операцій, що особливо важливо в умовах обробки великих обсягів даних із високим ступенем паралелізму. Метод *LU/QR*-факторизації, незважаючи на дещо більший час виконання, виявив високу стабільність і універсальність при роботі зі складною структурою запитів. У той же час метод *SVD*, хоча і забезпечив цінну інформаційну компресію, вимагає значного обсягу пам'яті, що може обмежувати його використання в режимах реального часу. Методи лінійних рівнянь для селекції показали збалансованість між швидкістю та ресурсною помірністю, зокрема при обробці умов із високою логічною складністю.

Таким чином, інтеграція методів лінійної алгебри в СУБД здатна суттєво підвищити ефективність обробки складних запитів. Матричне подання даних забезпечує можливість формалізувати реляційні операції та виконувати їх із використанням високопродуктивних обчислень, включно з GPU-прискоренням.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

У дослідженні обґрунтовано ефективність застосування методів лінійної алгебри для оптимізації запитів у СУБД. Формалізація табличних структур у вигляді матриць дозволяє трансформувати реляційні операції в обчислювально ефективні матричні процедури, що особливо актуально в умовах паралельної та GPU-архітектури.

Результати експериментального моделювання демонструють доцільність використання різних лінійно-алгебраїчних методів залежно від типу запиту та обчислювального середовища. Зокрема, матричне моделювання *JOIN*-операцій виявилось найшвидшим за часом виконання, тоді як *LU/QR*-факторизація забезпечила стабільну продуктивність у складних запитах з багаторівневою логікою. Метод *SVD* підтвердив свою ефективність для задач тематичного групування та кешування, хоча і потребує значних ресурсів пам'яті. Використання систем лінійних рівнянь для фільтрації забезпечило баланс між швидкістю й точністю при реалізації умов селекції.

Таким чином, застосування лінійної алгебри відкриває перспективні напрями розвитку архітектури СУБД, зокрема для гібридних або паралельних систем, де поєднується обробка структурованих та напівструктурованих даних із високопродуктивними числовими обчисленнями.

У подальших дослідженнях доцільно розширити модель на графові структури та об'єднати її з алгоритмами машинного навчання, зокрема для предиктивного кешування та автоматичного перепланування запитів.

Література

1. Суліма С.В., Єрмолаєв О.Д. Метод оптимізації SQL запитів системи управління базами даних. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2023. Т. 2, № 72. С. 151–157. DOI: <https://doi.org/10.26906/sunz.2023.2.151>.
2. Dudar Z., Liashyk V. Method for solving quantifier linear equations based on the algebra of linear predicate operations. *Radioelectronic and Computer Systems*. 2025. Vol. 2025, № 1. P. 102–112. DOI: <https://doi.org/10.32620/reks.2025.1.07>.
3. Sun W., Katsifodimos A., Hai R. Accelerating machine learning queries with linear algebra query processing. *Distributed and Parallel Databases*. 2025. Vol. 43, no. 1. DOI: <https://doi.org/10.1007/s10619-024-07451-7>.
4. Langerak A.J., Frasinca F., Klinkhamer J. JAL: an algebra for JSON query optimization. *World Wide Web*. 2025. Vol. 28, no. 3. DOI: <https://doi.org/10.1007/s11280-025-01336-0>.
5. Зінченко П., Мехед Д. (2025). Оптимізація запитів GraphQL. *Технічні науки та технології*. 2025. № 1 (39). С. 161–169. [https://doi.org/10.25140/2411-5363-2025-1\(39\)-161-169](https://doi.org/10.25140/2411-5363-2025-1(39)-161-169).
6. Fent Ph., Moerkotte G., Neumann Th. Asymptotically better query optimization using indexed algebra. *Proceedings of the VLDB Endowment*. 2023. Vol. 16, no. 11. P. 3018–3030. DOI: <https://doi.org/10.14778/3611479.3611505>.
7. Gaikwad T., Mohite A., Tarawade M. Combinatorial approaches to database query optimization: enhancing

- efficiency and performance. *International Research Journal on Advanced Engineering and Management (IRJAEM)*. 2025. Vol. 3, no. 04. P. 1449–1452. DOI: <https://doi.org/10.47392/irjaem.2025.0235>.
8. Veerappa P. M., Chikkamannur A. A. Prime learning – ant colony optimization technique for query optimization in distributed database system. *International Journal of Engineering Trends and Technology*. 2022. Vol. 70, no. 8. P. 158–165. DOI: <https://doi.org/10.14445/22315381/IJETT-V70I8P216>.
9. Chatterjee N., Sharma M., Sood S., Shubneet, Yadav A.R. Database management systems. Intelligent shields: artificial intelligence and machine learning for cybersecurity. *IIP Series*. 2025. Vol. 4. P. 103–114. Mode of access: <https://iipseries.org/assets/docupload/rs120254AFBD804DED1AFA.pdf> (date of access: 20.07.2025).
10. Fahima M.M.F., Sahna Sreen A.H., Fathima Ruksana S.L., Weihena D.T.E., Majid M.H.M. Machine learning for database management and query optimization. *Elementaria: Journal of Educational Research*. 2024. Vol. 2, no. 1. P. 96–108. DOI: <https://doi.org/10.61166/elm.v2i1.66>.
11. Guo Y., Li G., Hu R., Wang Y. In-database query optimization on SQL with ML predicates. *The VLDB Journal*. 2024. Vol. 34, no. 1. DOI: <https://doi.org/10.1007/s00778-024-00888-3>.
12. Ma X. The analysis of the internet of things database query and optimization using deep learning network model. *PLOS ONE*. 2024. Vol. 19, no. 6. P. e0306291. DOI: <https://doi.org/10.1371/journal.pone.0306291>.
13. Семенюк В.В. Аналіз та оптимізація продуктивності баз даних у розподілених системах. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2024. № 54. С.199-205. DOI: <https://doi.org/10.36910/6775-2524-0560-2024-54-25>
14. Ільїна І.В., Кадубенко Д.С., Ільїн Д.О. Підвищення продуктивності систем управління базами даних. *Системи управління, навігації та зв'язку*. 2018. Вип. 1. С. 105–109. DOI: <https://doi.org/10.26906/sunz.2018.1.105>.
15. Белоус Р.В., Крылов Є.В. Оптимізація використання RAFT в розподілених системах з базами даних. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2024. Вип. 1. № 3. С. 37–41. DOI: <https://doi.org/10.32782/2663-5941/2024.3.1/07>.

References

1. Sulima, S., Iermolaiev, O. (2023). Metod optymizatsii SQL zapytiv systemy upravlinnia bazamy danykh [The method of optimizing SQL queries of the database management system]. *Control, Navigation and Communication Systems*, vol. 2, no. 72, pp. 151–157.
2. Dudar, Z., Liashyk, V. (2025). Method for solving quantifier linear equations based on the algebra of linear predicate operations. *Radioelectronic and Computer Systems*, vol. 1, pp. 102–112.
3. Sun, W., Katsifodimos, A., Hai, R. (2025). Accelerating machine learning queries with linear algebra query processing. *Distributed and Parallel Databases*, vol. 43, no. 1. DOI: <https://doi.org/10.1007/s10619-024-07451-7>.
4. Langerak, A.J., Frasinca, F., Klinkhamer, J. (2025). JAL: an algebra for JSON query optimization. *World Wide Web*, vol. 28, no. 3. DOI: <https://doi.org/10.1007/s11280-025-01336-0>.
5. Zinchenko, P., Mekhed, D. (2025). Optymizatsiia zapytiv GraphQL [Optimization of GraphQL queries]. *Technical Sciences and Technologies*, vol. 1 (39), pp. 161–169.
6. Fent, Ph., Moerkotte, G., Neumann, Th. (2023). Asymptotically better query optimization using indexed algebra. *Proceedings of the VLDB Endowment*, vol. 16, no. 11, pp. 3018–3030.
7. Gaikwad, T., Mohite, A., Tarawade, M. (2025). Combinatorial approaches to database query optimization: enhancing efficiency and performance. *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 3, no. 04, pp. 1449–1452.
8. Veerappa, P. M., Chikkamannur, A. A. (2022). Prime learning – ant colony optimization technique for query optimization in distributed database system. *International Journal of Engineering Trends and Technology*, vol. 70, no. 8, pp. 158–165.
9. Chatterjee, N., Sharma, M., Sood, S., Shubneet, Yadav, A.R. (2025). Database management systems. Intelligent shields: artificial intelligence and machine learning for cybersecurity. *IIP Series*, vol. 4, pp. 103–114, available at URL: <https://iipseries.org/assets/docupload/rs120254AFBD804DED1AFA.pdf>.
10. Fahima, M.M.F., Sahna Sreen, A.H., Fathima Ruksana, S.L., Weihena, D.T.E., Majid, M.H.M. (2024). Machine learning for database management and query optimization. *Elementaria: Journal of Educational Research*, vol. 2, no. 1, pp. 96–108.
11. Guo, Y., Li, G., Hu, R., Wang, Y. (2024). In-database query optimization on SQL with ML predicates. *The VLDB Journal*, vol. 34, no. 1. DOI: <https://doi.org/10.1007/s00778-024-00888-3>.
12. Ma, X. (2024). The analysis of the internet of things database query and optimization using deep learning network model. *PLOS ONE*, vol. 19, no. 6, pp. e0306291.
13. Semeniuk, V. (2024). Analiz ta optymizatsiia produktyvnosti baz danykh u rozpodilenykh systemakh [Analysis and optimization of database performance in distributed systems]. *Computer-integrated technologies: education, science, production*, vol. 54, pp. 199–205.
14. Ilina, I.V., Kadubenko, D.S., Ilin D.O. (2018). Pidvyshchennia produktyvnosti system upravlinnia bazamy danykh [Improving the performance of database management systems]. *Sistemi upravlinnia, navigacii ta zv'язku*, vol. 1, pp. 105–109.
15. Belous, R.V., Krylov, Ye.V. (2024). Optymizatsiia vykorystannia RAFT v rozpodilenykh systemakh z bazamy danykh [Optimization of raft usage in distributed systems with databases]. *Scientific notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences*. vol. 1, no. 3, pp. 37–41.