

Соловей Ольга Леонідівна

Кандидат технічних наук, докторантка кафедри інформаційних технологій,

<https://orcid.org/0000-0001-8774-7243>

Київський національний університет будівництва і архітектури, Київ

**СУЧАСНІ ТЕХНОЛОГІЧНІ РІШЕННЯ ЗБЕРІГАННЯ ДАНИХ
У ПРОЄКТАХ МІСЬКОГО БУДІВНИЦТВА**

Анотація. Цифровізація будівельної галузі генерує величезні обсяги даних, що створює виклик для вибору оптимальної архітектури сховища. У статті проводиться порівняльний аналіз продуктивності трьох класів сучасних технологій зберігання даних: традиційної реляційної SQL-бази даних, глобально розподіленої NoSQL-бази даних (Cosmos DB) та технології «Озеро Даних» (Delta Lake). Метою є визначення їхньої придатності як сховища даних для проєктів міського будівництва. Включення до аналізу цих трьох технологій дозволяє виконати порівняльний аналіз сучасних рішень зберігання даних (Cosmos DB та Delta Lake) із традиційним рішенням для будівельних проєктів – SQL DB. Для цього було проведено кількісні тести, що імітують операції запису, читання та агрегації на наборах даних, обсяг яких варіювався від 10 тис. до 100 тис. рядків. Вимірювався час виконання для трьох типів навантажень: масовий запис, повне сканування та аналітична агрегація. Для оцінки використовується формальна модель, яка враховує як абсолютний час виконання (f_1), так і коефіцієнт масштабованості (f_2), що характеризує зміну продуктивності зі збільшенням обсягу даних. Результати показали, що жодна з технологій не є універсальним рішенням. Реляційна SQL-база даних демонструє найнижчу затримку для операційних запитів (час агрегації залишається мінімальним, зростаючи помірно від 10 мс до 110 мс). Водночас, Delta Lake показує найнижчі значення $f_1(t)$ та $f_2(t)$ для навантажень типу O1 ($f_1(t)$ має значення 3.47 мс при 10 тис. рядків до 32 мс при 100 тис. рядків; $f_2(t)$ має середнє значення ≈ 0.0004). Cosmos DB демонструє найбільші значення $f_1(t)$ для навантаження типу O1 – від 42 000 мс до 1 020 000 мс при 100 тис. рядків. На основі отриманих емпіричних даних обґрунтовано висновок про необхідність впровадження гібридної дворівневої архітектури, де SQL-база даних використовується для операційного рівня (OLTP), а технологія Delta Lake – для аналітичного рівня (OLAP). Така архітектура дозволить ефективно вирішувати як поточні, так і майбутні завдання з управління даними в будівництві. Робота спрямована на розробку гібридної дворівневої архітектури сховища даних для системи управління даними проєктів міського будівництва.

Ключові слова: великі дані; технології зберігання даних; швидкодія операційного рівня; швидкодія аналітичного рівня; масштабованість

Вступ

Цифрова трансформація галузі архітектури, інженерії та будівництва призвела до експоненційного зростання обсягу даних, що суттєво змінює парадигми управління ними, дозволяючи впровадження підходів до прийняття рішень на основі аналізу даних [1; 2].

Екосистема даних сучасного міського будівельного проєкту, яка охоплює все, від моделювання інформації будівель (BIM) і фотограмметричних сканувань до даних сенсорів Інтернету речей (IoT) з обладнання та реальних фінансових транзакцій, підпадає під визначення "Великі дані". Це визначення не обмежується лише розміром даних, а виправдане їх відповідністю п'яти основним вимірам (5V) моделі "Великих даних" [3]:

Обсяг (Volume): Один великий будівельний проєкт генерує терабайти даних, що надходять від високоякісних 3D моделей, зображень, отриманих за допомогою дронів, великої кількості документації проєкту та безперервних потоків даних з обладнання на майданчику.

Швидкість (Velocity): Сенсори IoT на будівельному обладнанні безперервно передають оперативні дані, в той час як платформи для управління проєктами отримують постійні оновлення про завдання, ресурси та хід виконання, що вимагає негайної обробки та аналізу.

Різноманіття (Variety): Дані є гетерогенними, варіюючись від високо структурованих даних у реляційних базах даних (витрати та графіки) до напівструктурованих даних (журнали сенсорів у форматах JSON/XML) і неструктурованих даних

(архітектурні креслення, юридичні контракти та відеофутаж з камер на майданчику) [4].

Достовірність (Veracity): Забезпечення точності, повноти та надійності даних, що є критично важливим для правильного аналізу та прийняття рішень, оскільки неточні чи неповні дані можуть призвести до помилкових висновків і негативних наслідків для проєктів.

Цінність (Value): Дані надають можливість для оптимізації розподілу ресурсів, прогнозного обслуговування та підвищення прибутковості проєктів.

Основні проблеми управління "Великими даними", з якими стикається галузь сьогодні, пов'язані з необхідністю створення інфраструктури даних, яка одночасно є надійною, масштабованою та ефективною. До цих проблем належать [5]:

Надійне та стабільне зберігання даних: Втрати чи пошкодження даних може призвести до значних фінансових та юридичних наслідків. Традиційні односерверні бази даних часто не мають необхідної стійкості до збоїв для критичних операцій.

Масштабованість: Рішення для зберігання повинно мати можливість масштабуватися, щоб задовольнити зростаючий обсяг та швидкість даних протягом життєвого циклу проєкту та для кількох одночасних проєктів без зниження продуктивності.

Ефективна обробка даних: Інфраструктура повинна підтримувати різноманітні робочі навантаження – від швидких транзакційних запитів до складних аналітичних запитів. Вибір архітектури зберігання безпосередньо впливає на затримку запитів та можливість застосування передових аналітичних методів і машинного навчання.

Подолання цих проблем є важливим для впровадження підходів до управління даними у галузі міського будівництва.

Мета статті

Ця стаття має на меті оцінити та порівняти три різні класи сучасних рішень для зберігання даних - традиційну реляційну базу даних SQL, глобально розподілену NoSQL базу даних (Cosmos DB) та технологію «озера даних» (Delta Lake), а також надати рекомендації щодо побудови архітектури сховища даних для системи управління даними проєктів міського будівництва.

Аналіз останніх досліджень і публікацій

Принцип розділення транзакційних (OLTP) та аналітичних (OLAP) навантажень є базовою концепцією в архітектурі даних, найбільш чітко сформульованою в таких підходах, як Lambda-архітектура [6], а також у традиційних методологіях побудови сховищ даних.

База даних SQL традиційно обирається як операційне ядро платформи даних будівельних проєктів. Реляційна база даних SQL реалізує стратегію централізованого сховища даних, що зберігає структуровані дані, отримані з різних джерел, у формі, оптимізований для запитів та аналізу [7]. Такі сховища використовують процес обробки даних під назвою «Вилучення, перетворення, завантаження» (ETL). В цьому процесі дані отримуються з вихідних систем, перетворюються у формат, придатний для аналізу, а потім завантажуються в сховище. Процес ETL орієнтований на схему, спрямовану на оптимізацію запитів та забезпечення узгодженості даних, що робить архітектуру сховища даних більш придатною для складного аналізу та звітності [7].

Технологія «Озеро даних» (Delta Lake) – це сучасна технологія сховища великих даних, створена для моделювання архітектури, необхідної для сучасної бізнес-аналітики, звітності та машинного навчання [8]. Масштабованість «озера даних» досягається завдяки тому, що дані зберігаються у розподілених файлових системах (HDFS, Amazon S3, Azure Blob Storage), які автоматично масштабуються горизонтально (шляхом додавання вузлів), паралельній обробці, незалежному масштабуванню обчислень і сховища, а також гнучкій схемі доступу до даних, що дозволяє ефективно працювати з будь-якими обсягами та типами даних [9]. Озера даних використовують процес під назвою «Вилучення, завантаження та перетворення» (ELT), у якому дані надходять в озеро в необробленому форматі, а потім перетворюються, коли це необхідно для аналізу. Ця стратегія надає можливість застосовувати різні методи обробки даних та інструменти аналітики до даних, що зберігаються в озері, без необхідності повного перетворення даних [10].

Особливості Cosmos DB безпосередньо відповідають кільком зростаючим і критично важливим вимогам великих міських будівельних проєктів, а саме: глобальній доступності даних у режимі реального часу, високій масштабованості без втрати продуктивності, мінімальним затримкам при обробці запитів, а також підтримці різноманітних моделей даних (документна, ключ-значення, графова та колонкова). Ця база даних забезпечує гнучку схему NoSQL документної бази даних, що спрощує імпорту і інтеграцію різних джерел даних порівняно з жорсткою реляційною схемою [11].

У дослідженні [12] виконано емпіричне порівняння технології NoSQL (Cosmos DB) та реляційної бази даних (SQL) за критеріями: пропускну здатність і масштабованість. Визначено, що реляційна база даних (SQL) є ефективною технологією для складних запитів та сильною узгодженості. Cosmos DB рекомендується для

масштабованих, глобально розподілених програм з простими шаблонами запитів та гнучкими моделями даних.

Таким чином, дослідження [12] орієнтоване на аналіз ефективності технологій з точки зору OLTP ефективності і не включає оцінювання технології для аналітичних навантажень.

Дослідження [13] виконало емпіричне порівняння двох провідних парадигм сучасної аналітики даних: традиційного хмарного сховища даних (DW) та нового Data Lakehouse (DLH). Ефективність кожного сховища визначається на основі OLAP тестів, які включають складні SQL-запити та паралельні запити. Також, DW та DLH оцінюються за їхніми ключовими нефункціональними атрибутами.

Таким чином, дослідження [13] включає рекомендації щодо вибору аналітичного механізму, залишаючи без уваги питання визначення технології для реалізації сховища даних, яке має бути в основі визначеного аналітичного механізму.

Метою дослідження [14] є визначення продуктивності форматів таблиць Delta Lake, Apache Iceberg, Apache Hudi, які лежать в основі Delta Lakehouse. Оцінювання виконується за результатами тестів на ефективність виконання транзакційних навантажень, таких як: додавання чи зміна значень даних, зміна типу даних, виконання запитів до історичних версій даних. Також проводилось тестування того, як кожен формат працює під час керування даними у форматі JSON, коли ці дані отримуються з приладів Інтернету речей. За результатами дослідження [14] визначено ефективність формату таблиць Delta Lake, які будуть використовуватись.

Аналіз сучасних досліджень показав, що на сьогодні немає кількісного порівняння ефективності технологій зберігання даних SQL, Cosmos DB та Delta Lake. Тому це дослідження є актуальним.

Виклад основного результату

Нехай T – це множина, яка включає технології SQL DB, Cosmos DB та Delta Lake, визначені для порівня в межах даного дослідження. Тоді, функцію загальної оцінки кожної $t \in T$ опишемо як суму двох критеріїв:

$$F(t) = f_1(t) + f_2(t), \quad (1)$$

де $f_1(t)$ – час, необхідний для виконання запиту на технології $t \in T$; $f_2(t)$ – масштабованість технології $t \in T$, що вимірюється через зміну часу виконання запиту в залежності від об'єму даних.

Для двох основних парадигм обробки даних, поширених у сучасних архітектурах: онлайн транзакційної обробки (OLTP) та онлайн аналітичної обробки (OLAP) визначимо множину O , яка включає

три основні типи навантажень: O_1 – масова вставка даних (BULK INSERT): пакетне завантаження даних у відповідну систему зберігання; O_2 – вибірка всіх даних (SELECT * FROM): отримання всього набору даних шляхом повного сканування таблиці або еквівалентного запиту; O_3 – агрегація (SELECT AVG(1) GROUP BY): виконання агрегатного запиту, що відповідає типовому аналітичному навантаженню.

Функцію $f_1(t)$ для кожного типу навантаження $o \in O$ опишемо виразами:

$$f_1^{O_1}(t) = c_1 \cdot N, \quad (2)$$

$$f_1^{O_2}(t) = c_2 \cdot N, \quad (3)$$

$$f_1^{O_3}(t) = c_3 \cdot N \cdot \log(N), \quad (4)$$

де c_1, c_2, c_3 – константи, які залежать від того, як швидко технологія $t \in T$ може виконувати тип навантаження $o \in O$; N – кількість рядків у таблиці.

Функцію $f_2(t)$ для навантажень опишемо лінійною функцією:

$$f_2^{O_i}(t) = \frac{f_1^{O_i(T) \cdot (N_2)} - f_1^{O_i(T) \cdot (N_1)}}{N_2 - N_1}, \quad (5)$$

де N_1, N_2 – два різні об'єми даних, для яких проводиться вимірювання, $i \in \{1, 2, 3\}$.

Для забезпечення порівняння обраних технологій зберігання даних було розроблено експеримент. Основна задача – емпірично оцінити продуктивність систем на основі функцій (2–5) з типів навантажень $t \in T$.

У межах експерименту згенеровано синтетичний набір даних у форматі JSON, який імітує типові напівструктуровані дані, властиві сучасним аналітичним сценаріям. Обсяг даних варіювався в межах від 10 000 до 100 000 рядків, з кроком у 10 000 записів на кожну ітерацію.

Для кожного типу запиту та обсягу даних тест для обчислення $f_1(t)$ виконувався п'ять разів. Перше виконання ігнорувалося з метою виключення ефекту "холодного старту" або компіляції в реальному часі. Остаточне значення $f_1(t)$ є середнє значення часу виконання за наступні чотири запуски.

Система працювала під управлінням 64-бітної Windows 10, із 16 ГБ оперативної пам'яті, процесором Intel Core i7 та твердотільним накопичувачем (SSD). Для Cosmos DB використовувався офіційний емулятор, розгорнутий локально за допомогою Docker, що забезпечувало поведінку, максимально наближену до продуктивного середовища.

Результати проведених експериментів, представлені на рис. 1, 2, показують різні профілі продуктивності для кожної технології зберігання даних, що підкреслює, що їхня придатність сильно залежить від конкретного робочого навантаження та обсягу даних.

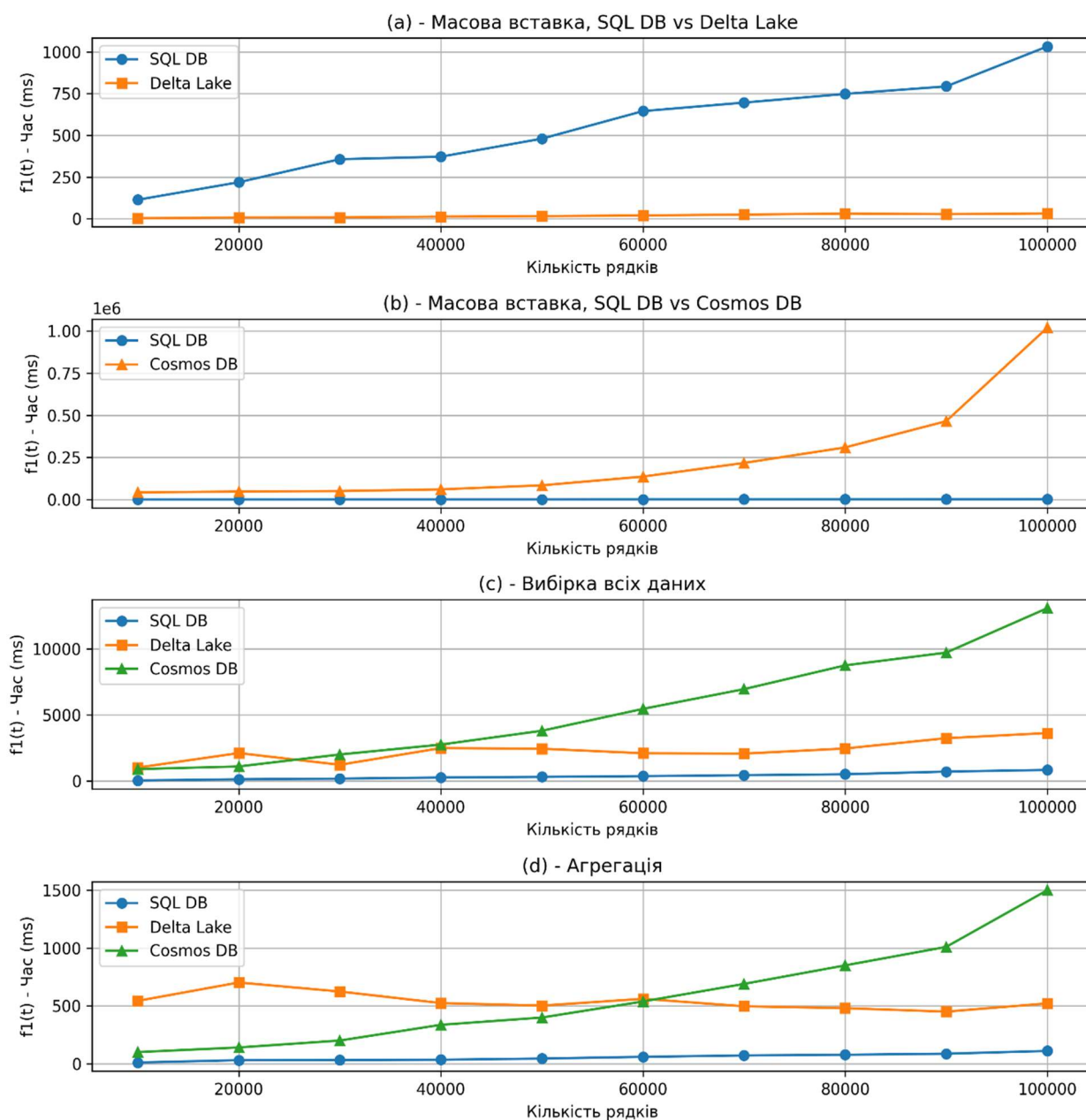


Рисунок 1 – Час, необхідний для виконання запиту на технології – SQL DB, Delta Lake, Cosmos DB

SQL демонструє стабільну поведінку, для навантаження типу O_1 та O_2 , із значеннями $f_1(t)$, що зростають лінійно – від 114 мс до 1032 мс (рис. 1, а), та від 30 мс до 827 мс (рис. 1, с). Час агрегації залишається мінімальним, зростаючи помірно від 10 мс до 110 мс, що свідчить про ефективність SQL у виконанні обчислень над реляційними даними (рис. 1, d).

Delta Lake, оптимізована для роботи з великими даними на розподілених архітектурах, демонструє найнижчі значення $f_1(t)$ для навантаження типу O_1 (3.47 мс при 10 тис. рядків до 32 мс при 100 тис. рядків) (рис. 1, а), Однак вона має значно вищі значення $a_1(t)$ для навантажень типу O_2 та O_3 (3624 мс

на читання та 521 мс на агрегацію при 100 тис. рядків) (рис. 1, с – d), що відображає накладні витрати на обробку формату даних та виконання через Spark.

Cosmos DB, показує найбільші значення $f_1(t)$ для навантаження типу O_1 від 42000 мс до 1020000 мс при 100 тис. рядків (рис. 1, а), імовірно, через внутрішні механізми забезпечення узгодженості, індексування та розподілену архітектуру. Хоча навантаження типу O_3 в Cosmos DB демонструють кращі показники, ніж у Delta Lake для кількості рядків до 600, вони зростають більше 1400 мс для 100 тис. рядків, що є найбільшим значенням часу порівняно з технологіями SQL DB і Delta Lake.

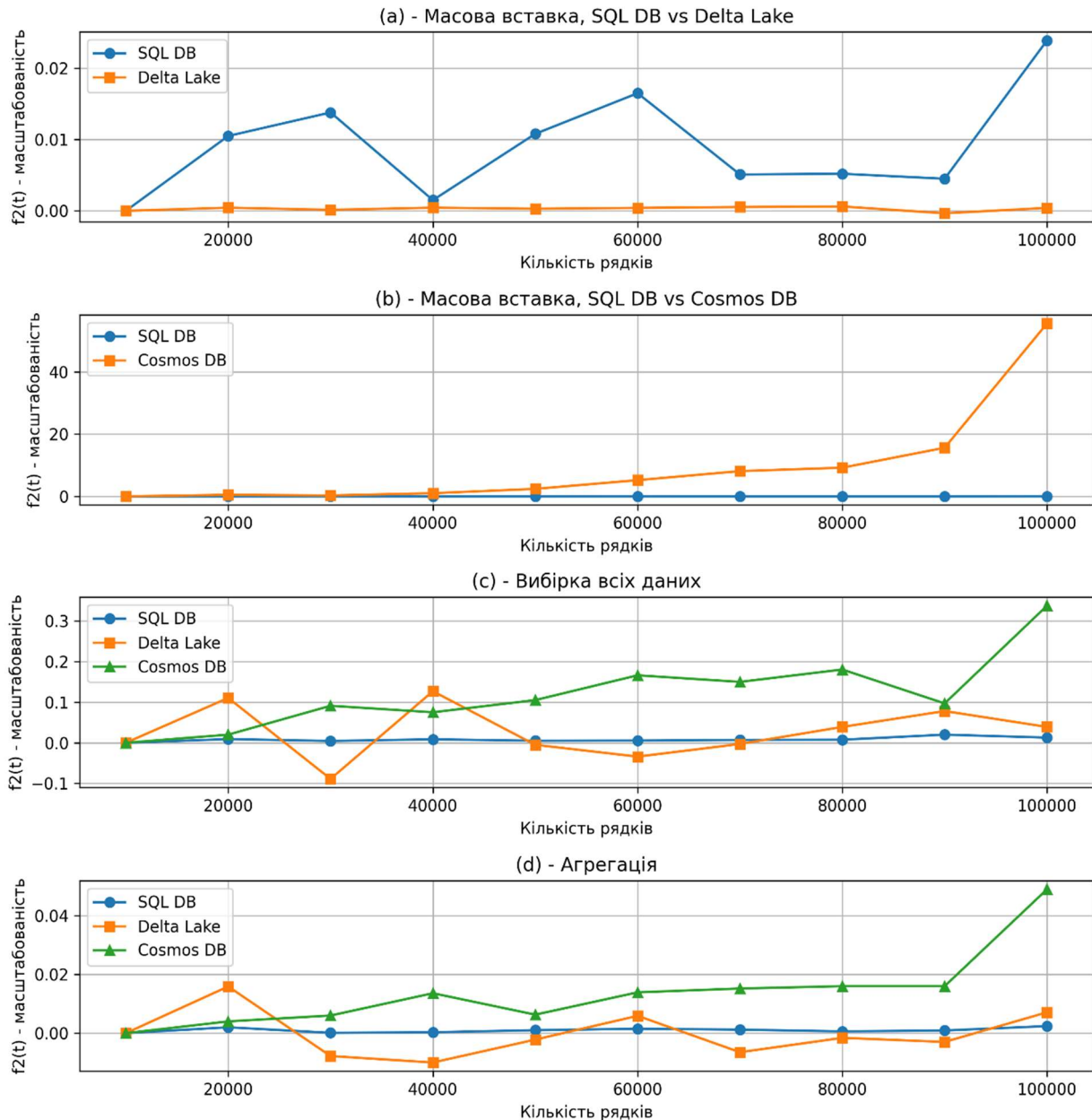


Рисунок 2 – Масштабованість технології залежно від кількості записів

Аналіз графіків функції масштабованості $f_2(t)$ на рис. 2 вказує, що для навантаження O_1 технологія Delta Lake має найнижчі значення $f_2(t)$ (середнє значення ~ 0.0004 , близьким до нуля) (рис. 2, а). Це експериментально підтверджує, що система є надмасштабованою та оптимізованою для запису. Її продуктивність майже не залежить від обсягу даних для цього завдання. Технологія SQL показує низьке, але стабільно позитивне значення f_2 (середнє значення ~ 0.01), із витратним ефектом для кожного додаткового рядка (рис. 2, а). Функція $f_2(t)$ (Cosmos DB) прискорюється (від 0.5 до 55.5), що свідчить про немасштабованість для навантаження O_1 , оскільки накладні витрати його розподіленої архітектури збільшуються із зростанням обсягу (рис. 2, б).

Для навантаження O_2 технологія SQL показує значення, близьке до нуля (середнє значення ~ 0.009) незалежно від кількості записів (рис. 2, с). f_2 для Cosmos DB (середнє значення ~ 0.15) вищий, ніж у SQL. Це підтверджує, що продуктивність читання безпосередньо залежить від обсягу даних, які передаються через мережу. $f_2(t)$ для Delta Lake непослідовне, показуючи як позитивні, так і негативні значення. Це вказує на те, що продуктивність технології для навантаження типу O_2 є нелінійною функцією, яка залежить від розміру даних.

Для навантаження O_3 функція f_2 для Delta Lake приймає від'ємні значення в 67% випадках, що означає, що після того, як виконані функції

ініціалізації Spark, час для виконання агрегації майже не залежить від розміру даних. Функція $f_2(t)$ для SQL приймає значення (від 0 до 0.0024). Це вказує на те, що технологія обробляє операції майже лінійно. Функція $f_2(t)$ для Cosmos DB приймає позитивні і зростаючі значення, що підкріплює те, що система не оптимізована для аналітичної масштабованості.

Висновки

Результати експериментів, представлені на рис. 1, 2 показують, що вибір технології збереження даних залежить від типу навантаження та обсягу даних. SQL демонструє стабільну продуктивність для операцій O_1 та O_2 , де час виконання зростає лінійно, що робить його ефективним для транзакційних навантажень. Проте при великих обсягах даних продуктивність SQL знижується, що робить його менш оптимальним для обробки великих масивів даних.

Delta Lake, оптимізована для роботи з великими даними, продемонструвала низькі значення $f_1(t)$ для навантаження O_1 , але для O_2 та O_3 спостерігалися значні накладні витрати, що знижує її ефективність при складних запитах та агрегаціях. Cosmos DB показала високу затримку при операціях типу O_1 через внутрішні механізми забезпечення узгодженості та

розподілену архітектуру, а для O_3 її продуктивність була найгіршою серед досліджених технологій.

Аналіз масштабованості $f_2(t)$ підтверджує, що Delta Lake є найбільш ефективною для запису великих даних та агрегацій на великих обсягах даних. SQL показує стабільну масштабованість для навантаження O_2 , тоді як Cosmos DB демонструє зростаючі затримки при збільшенні обсягу даних.

Зважаючи на отримані результати, жодна з технологій не є універсальною для всіх типів навантажень, характерних для проєктів міського будівництва. Найкращим рішенням є гібридна дворівнева архітектура даних, яка поєднує сильні сторони кожної технології:

- Для операційного рівня (OLTP) рекомендується використання SQL, який забезпечує найкращу продуктивність для транзакційних операцій.

- Для аналітичного рівня (OLAP) оптимальним вибором є Delta Lake, яка показала високу ефективність при запису великих даних і масштабованості при агрегаціях.

Наступні наукові дослідження будуть спрямовані на розробку гібридної дворівневої архітектури сховища даних для системи управління даними проєктів міського будівництва.

Список літератури / References

1. Honcharenko, T., Khrolenko, V., Gorbatiuk, I., Liashchenko, M., Bodnar, N., & Sherif, N. H. (2024). Smart integration of information technologies for city digital twins. In *2024 35th Conference of Open Innovations Association (FRUCT)* (pp. 253-258). IEEE.
2. Solovei, O., Honcharenko, T., & Fesan, A. (2024). Technologies to manage big data of urban building projects. *Management of Development of Complex Systems*, 60, 121–128. URL: <https://doi.org/10.32347/2412-9933.2024.60.121-128>.
3. Dolhopolov, S., Honcharenko, T., Savenko, V., Balina, O., Bezklubenko, I., & Liashchenko, T. (2023). Construction site modeling objects using artificial intelligence and BIM technology: A multi-stage approach. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)* (pp. 174-179). IEEE.
4. Ohene, E., Nani, G., Antwi-Afari, M. F., Darko, A., Addai, L. A., & Horvey, E. (2024). Big data analytics in the AEC industry: scientometric review and synthesis of research activities. *Engineering, Construction and Architectural Management*. URL: <https://doi.org/10.1108/ECAM-01-2024-0144>.
5. Kulikov, P., Ryzhakova, G., Honcharenko, T., & Ryzhakov, D. (2020). Olap-tools for the formation of connected and diversified production and project management systems. *International Journal of Advanced Trends in Computer Science and Engineering*, 9 (5), 8670-8676.
6. Rana, M. (2025). A review of the impact of big data on smart cities. URL: <https://doi.org/10.56557/jobari/2025/v31i29162>.
7. Jamarani, A., Haddadi, S., Sarvizadeh, R., Haghi Kashani, M., Akbari, M., & Moradi, S. (2024). Big data and predictive analytics: A systematic review of applications. *Artificial Intelligence Review*, 57 (7), 176. URL: <https://doi.org/10.1007/s10462-024-10811-5>.
8. Hassan, I. (2024). Storage structures in the era of big data: from data warehouse to lakehouse. *Journal of Theoretical and Applied Information Technology*, 102 (6).
9. Azzabi, S., Alfughi, Z., & Ouda, A. (2024). Data lakes: A survey of concepts and architectures. *Computers*, 13 (7), 183. URL: <https://doi.org/10.3390/computers13070183>.
10. Schneider, J., Gröger, C., Lutsch, A., Schwarz, H., & Mitschang, B. (2024). The lakehouse: State of the art on concepts and technologies. *SN Computer Science*, 5 (5), 449.
11. Plazotta, M., & Klettke, M. (2024). Data architectures in cloud environments. *Datenbank-Spektrum*, 24 (3), 243–247. URL: <https://doi.org/10.1007/s13222-024-00490-5>.

12. Panda, S. P. (2024). Comparative analysis of azure cosmos DB vs. traditional RDBMS on cloud. *Traditional RDBMS on Cloud* (July 22, 2024). URL: <https://doi.org/10.5281/zenodo.15481723>.
13. Salqvist, P. (2024). A comparative study of the data warehouse and data lakehouse architecture.
14. Eswararaj, D., Nellipudi, A. B., & Kollati, V. (2025). A comparative study of delta parquet, iceberg, and hudi for automotive data engineering use cases. *arXiv preprint arXiv:2508.13396*. URL: <https://doi.org/10.14445/23488387/IJCSE-V12I17P104>.

Стаття надійшла до редколегії 08.09.2025

Solovei Olga

Doctoral student of the department of information technologies, associate professor of the department of information technologies, <https://orcid.org/0000-0001-8774-7243>
Kyiv National University of Construction and Architecture, Kyiv

MODERN TECHNOLOGICAL DATA STORAGE SOLUTIONS IN URBAN CONSTRUCTION PROJECTS

Abstract. The digitalization of the construction industry generates vast amounts of data, posing a challenge in selecting an optimal storage architecture. This study presents a comparative performance evaluation of three classes of contemporary data storage technologies: the traditional relational SQL database, the globally distributed NoSQL database (Cosmos DB), and the Data Lake solution (Delta Lake). The objective was to assess their suitability as data repositories for urban construction projects. Quantitative experiments were conducted to simulate write, read, and aggregation operations on datasets ranging from 10,000 to 100,000 rows. Execution time was measured for three workload types: bulk insert, full scan, and analytical aggregation. A formal evaluation model was employed, incorporating both absolute execution time (f_1) and the scalability coefficient (f_2), which characterizes performance variation with increasing data volume. The results indicate that none of the evaluated technologies provides a universal solution. The relational SQL database exhibited the lowest latency for operational queries, with aggregation times remaining minimal and rising moderately from 10 ms to 110 ms. Delta Lake achieved the most favorable $f_1(t)$ and $f_2(t)$ values for OI-type workloads ($f_1(t)$ increasing from 3.47 ms at 10,000 rows to 32 ms at 100,000 rows; mean $f_2(t) \approx 0.0004$). By contrast, Cosmos DB showed the highest $f_1(t)$ values for OI-type workloads, ranging from 42,000 ms to 1,020,000 ms at 100,000 rows. Based on these findings, a hybrid two-tier architecture is proposed, in which the SQL database is employed at the operational layer (OLTP), while Delta Lake is adopted at the analytical layer (OLAP). This approach provides an effective balance for addressing both current and future data management challenges in the construction domain. Future work will focus on the design and implementation of such a hybrid architecture for urban construction data management systems.

Keywords: Big Data; Data Storage Technologies; Operational Tier Performance; Analytical Tier Performance; Scalability

Посилання на публікацію

- | | |
|------|---|
| APA | Solovei O. (2025). Modern technological data storage solutions in urban construction projects. <i>Management of Development of Complex Systems</i> , 63, 167–173, dx.doi.org/10.32347/2412-9933.2025.63.167-173 . |
| ДСТУ | Соловей О. Л. Сучасні технологічні рішення зберігання даних у проєктах міського будівництва. <i>Управління розвитком складних систем</i> . Київ, 2025. № 63. С. 167 – 173, dx.doi.org/10.32347/2412-9933.2025.63.167-173 . |