

Ратушний Дарій Миколайович

Аспірант.

НУБіП, Київ, Україна.

ORCID: 0009-0009-4699-9909

d.ratushnyi@nubip.edu.ua

Руденський Роман Анатолійович

Доктор економічних наук, професор.

НУБіП, Київ, Україна.

roman.rudensky@nubip.edu.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОДУКТИВНОСТІ ТА БЕЗПЕКИ ТИПІВ ORM SEQUELIZE ТА PRISMA У СУЧАСНИХ NODE.JS-СИСТЕМАХ

Анотація. У статті досліджується архітектурна проблема вибору інструменту об'єктно-реляційного відображення (ORM) для Node.js-систем, від якого безпосередньо залежать ефективність та масштабованість додатку. Робота фокусується на порівняльному аналізі двох провідних інструментів: Sequelize (v6.35.2), зрілої ORM на патерні Active Record, та Prisma (v5.7.0), сучасної ORM нового покоління з підходом "Schema-First" та потужною генерацією типів. **Метою роботи** є комплексний аналіз Sequelize та Prisma за двома напрямками: теоретичний аналіз підходів до проектування схеми даних та забезпечення безпеки типів у TypeScript, та експериментальне дослідження продуктивності для типових CRUD-операцій і складних аналітичних запитів. Для досягнення мети було розроблено методику дослідження: створено тестовий стенд на базі Node.js, PostgreSQL (v15) та Docker. Тестове середовище наповнене 1000 сутностями "користувач" та 5000 сутностями "пост". Бенчмаркінг проводився протягом 10 секунд з рівнем конкурентності 10 одночасних операцій. Було протестовано шість типових сценаріїв, включаючи просту вибірку (findAllUsers), фільтрацію (findUsersByAge), операцію з JOIN (findUserWithPosts), базові CRUD-операції (createUser, updateUser) та складний запит (complexQuery з JOIN, WHERE, ORDER BY та LIMIT). Отримані результати продемонстрували повну перевагу Prisma у всіх шести категоріях. Загальний середній показник запитів в секунду (RPS) для Prisma склав 1707.10, що на 50.8% вище, ніж у Sequelize (1131.96 RPS). Найбільш критична різниця зафіксована у продуктивності складного запиту (complexQuery): Prisma показала стабільний результат у 699.54 RPS, тоді як Sequelize показала лише 0.37 RPS. Це свідчить про перевагу Prisma у 188,964.9% у складних сценаріях, що пояснюється архітектурними відмінностями (використанням компільованого бінарного рушія запитів Prisma на Rust). У висновках сформульовано практичні рекомендації: Prisma однозначно рекомендується для нових, особливо TypeScript-проектів, завдяки неперевершеній безпеці типів та значно вищій продуктивності. Sequelize залишається вибором для існуючих проектів, але вимагає готовності до суттєвих накладних витрат на продуктивність у складних сценаріях.

Ключові слова: ORM; Sequelize; Prisma; Node.js; TypeScript; проектування баз даних; продуктивність; бенчмарк; обробка даних; RPS.

1. ВСТУП

Постановка проблеми. У сучасних програмних системах, побудованих на платформі Node.js, ефективна взаємодія з базами даних є наріжним каменем. Інструменти об'єктно-реляційного відображення (ORM) виступають критично важливим шаром абстракції, що перетворює об'єкти мови програмування (наприклад, JavaScript або TypeScript) у реляційні структури даних (таблиці, рядки) і навпаки. Вибір ORM є фундаментальним архітектурним рішенням, що безпосередньо впливає на швидкість розробки, легкість підтримки коду, безпеку типів та загальну продуктивність системи.

Аналіз останніх досліджень і публікацій. Існує значна кількість ресурсів, що описують переваги та недоліки окремих ORM. Офіційна документація Sequelize [1] та Prisma [2] детально розкриває їхні архітектурні підходи. Численні публікації у спільнотах розробників [3] часто фокусуються на синтаксичному порівнянні або

суб'єктивному "досвіді розробника" (Developer Experience). Однак, бракує актуальних, комплексних досліджень, які б одночасно порівнювали ці два інструменти за критеріями безпеки типів у TypeScript та вимірюваної продуктивності в контрольованих умовах. Невирішеною частиною проблеми залишається надання розробникам чітких, заснованих на даних рекомендацій щодо вибору ORM для конкретних практичних завдань проектування баз даних.

Мета. Проведення комплексного порівняльного аналізу ORM Sequelize та Prisma. Основні завдання дослідження: проаналізувати теоретичні основи та парадигми (Active Record vs. Schema-First), що лежать в основі проектування баз даних у Sequelize та Prisma; оцінити рівень забезпечення безпеки типів (type-safety) та його вплив на процес розробки програмних систем; розробити та провести експериментальне вимірювання продуктивності обох ORM для типових сценаріїв обробки даних (CRUD-операції та складні запити).

2. ТЕОРЕТИЧНІ ОСНОВИ

Sequelize: Парадигма "Code-First" та Active Record. Sequelize (v6.35.2) є однією з найстаріших та найзріліших ORM для Node.js. Вона побудована на патерні Active Record, де модель в коді є "розумною" і безпосередньо відповідає за зв'язок з таблицею в БД (включаючи методи .save(), .update(), .destroy()). Проектування схеми зазвичай відбувається за підходом "Code-First": розробник описує моделі та їхні зв'язки в коді, а потім інструмент sequelize-cli генерує файли міграцій для синхронізації БД. **Prisma: Парадигма "Schema-First" та генерація клієнта.** Prisma (v5.7.0) представляє нове покоління ORM. В її основі лежить підхід "Schema-First". Розробник визначає всю схему даних, включаючи моделі, зв'язки та обмеження, в одному декларативному файлі schema.prisma. Цей файл стає єдиним джерелом правди про структуру даних.

3. МЕТОДИ ДОСЛІДЖЕННЯ

Для тестування була використана реляційна схема з двох сутностей: таблиця User (1000 записів); таблиця Post (5000 записів). Оцінка продуктивності проводилась за двома ключовими показниками: RPS; середній час відповіді (мс). Вимірювання проводились для 6-ти операцій при тривалості тесту 10 секунд та конкурентності 10 одночасних операцій: findAllUsers; findUsersByAge; findUserWithPosts; createUser; updateUser; complexQuery - складний запит з JOIN, WHERE, ORDER BY та LIMIT.

4. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ.

Результати експериментального дослідження продуктивності.

Таблиця 1

Підсумкова таблиця порівняння RPS (Запитів в секунду)

Операція	Sequelize RPS	Prisma RPS	Переможець	Різниця
findAllUsers	642.49	1089.26	✓ Prisma	69.5%
findUsersByAge	1447.21	1756.15	✓ Prisma	21.3%
findUserWithPosts	1128.39	2512.79	✓ Prisma	122.7%
createUser	1777.64	2040.39	✓ Prisma	14.8%
updateUser	1795.64	2144.46	✓ Prisma	19.4%

complexQuery	0.37	699.54	✓ Prisma	188964.9%
Середнє	1131.96	1707.10	✓ Prisma	50.8%

Обговорення результатів таблиці. Результати, наведені у Таблиці 1, демонструють однозначну та повну перевагу Prisma у всіх вимірюваних категоріях. Prisma виявилася в середньому на 50.8% швидшою за загальним показником RPS. **Найбільш помітна різниця спостерігається у двох категоріях:** findUserWithPosts - Prisma показала на 122.7% вищу продуктивність. Це свідчить про те, що механізм "включення" пов'язаних даних у Prisma оптимізований значно краще, ніж аналогічний механізм у Sequelize, що часто призводить до проблеми "N+1" або менш ефективних SQL-запитів. complexQuery - цей тест виявив критичну проблему продуктивності Sequelize за умов конкурентного навантаження. **Аналіз вузьких місць.** Найбільш показовим результатом тестування стало вимірювання продуктивності complexQuery. Результати (див. Таблиця 1) виявили фактичну нездатність Sequelize впоратися з цим завданням під навантаженням. Prisma показала стабільний результат у 699.54 RPS із середнім часом відповіді 1.43 мс. Sequelize: Показала лише 0.37 RPS, при цьому середній час виконання одного запиту склав 2707.4 мс (понад 2.7 секунди). Різниця у 188,964.9% (або у 1890 разів) є колосальною.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ. Проведене дослідження дозволило сформулювати наступні висновки: підхід Prisma "Schema-First" є більш надійним для проектування баз даних, оскільки schema.prisma слугує єдиним джерелом правди. Підхід Sequelize "Code-First" пропонує більшу гнучкість, але підвищує ризик розсинхронізації коду та стану БД. Prisma забезпечує практично 100% безпеку типів "з коробки" завдяки генерації клієнта. Це є вирішальною перевагою для проектів на TypeScript, оскільки кардинально знижує кількість помилок під час обробки даних. Продуктивність: Експериментальні виміри чітко продемонстрували перевагу Prisma у продуктивності в усіх 6 тестових сценаріях. Prisma в середньому на 50.8% швидша за Sequelize. Особливо критична перевага (у 188,964.9%) спостерігається у складних запитах, що робить Prisma значно надійнішим вибором для систем з високим навантаженням та складною логікою обробки даних. **Перспективи подальших досліджень.** Подальші дослідження можуть бути спрямовані на аналіз продуктивності при роботі з нереляційними базами даних (зокрема, MongoDB, яку підтримують обидві ORM), а також тестування під значно вищими навантаженнями та рівнем конкурентності (наприклад, 100 або 1000 одночасних операцій) для виявлення "вузьких місць" у пулах з'єднань.

ПОСИЛАННЯ

- [1] "Sequelize documentation," *Sequelize.org*, 2025. [Online]. Available: <https://sequelize.org/docs/v6/>. [Accessed: Nov. 9, 2025].
- [2] "Prisma documentation," *Prisma.io*, 2025. [Online]. Available: <https://www.prisma.io/docs>. [Accessed: Nov. 9, 2025].
- [3] M. Zonov, "Choosing a Node.js ORM: Prisma vs. Sequelize in 2024," *LogRocket Blog*, Apr. 2, 2024.

MINISTRY OF EDUCATION
AND SCIENCE OF UKRAINE

NATIONAL UNIVERSITY
OF LIFE AND ENVIRONMENTAL
SCIENCES OF UKRAINE

FACULTY OF INFORMATION
TECHNOLOGY

МІНІСТЕРСТВО ОСВІТИ
І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

PROCEEDINGS

XIII International scientific
and practical conference

**GLOBAL AND
REGIONAL PROBLEMS OF
INFORMATIZATION IN
SOCIETY AND
NATURE USING
'2025**

13-14 November 2025

Kyiv, NULES of Ukraine

Kyiv 2025

МАТЕРІАЛИ

XIII Міжнародної науково-
практичної конференції

**ГЛОБАЛЬНІ ТА
РЕГІОНАЛЬНІ ПРОБЛЕМИ
ІНФОРМАТИЗАЦІЇ В
СУСПІЛЬСТВІ І
ПРИРОДОКОРИСТУВАННІ
'2025**

13-14 листопада 2025 року

Київ, НУБіП України

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАТЕРІАЛИ

ХІІІ Міжнародної науково-практичної конференції

ГЛОБАЛЬНІ ТА РЕГІОНАЛЬНІ ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ В СУСПІЛЬСТВІ І ПРИРОДОКОРИСТУВАННІ '2025

13-14 листопада 2025 року

Київ, НУБіП України

Київ 2025

УДК 004

Рекомендовано до друку вченою радою факультету інформаційних технологій Національного університету біоресурсів і природокористування України (протокол № 4 від 18.12.2025).

Укладач: д.т.н., доцент Шкарупило В.В.

Збірник матеріалів XIII Міжнародної науково-практичної конференції "Глобальні та регіональні проблеми інформатизації в суспільстві і природокористуванні '2025", 13–14 листопада 2025 року, НУБіП України, Київ. – К.: НУБіП України, 2025. – 206 с.

Відповідальність за зміст публікацій несуть автори.

© Національний університет біоресурсів
і природокористування України, 2025