

## ДОСЛІДЖЕННЯ ТА РОЗРОБКА ORM-СИСТЕМИ МОВОЮ ПРОГРАМУВАННЯ JAVA

Владислав Невірець

RESEARCH AND DEVELOPMENT OF THE ORM-SYSTEM IN JAVA PROGRAMMING

Vladyslav Nevirets

Волинський національний університет імені Лесі Українки, просп. Волі, 13, Луцьк,  
43025, Україна

**Abstract.** *This paper presents the development of a lightweight, custom Object-Relational Mapping (ORM) system in Java aimed at simplifying database operations for Java applications. The project resulted in a functional and extensible solution that provides essential CRUD operations, automatic mapping between Java classes and database tables. Designed and implemented entirely by a solo developer, the system demonstrates how core ORM functionality can be achieved without relying on external libraries such as Hibernate. The research explores common patterns in ORM design and offers a practical example of building a database abstraction layer from scratch using JDBC and reflection. This project validates the feasibility of constructing a tailored ORM for educational or lightweight production use cases.*

**Актуальність.** Об'єктно-реляційне відображення (ORM) [1] - це техніка, яка використовується для створення "моста" між об'єктно-орієнтованими програмами і, в більшості випадків, реляційними базами даних. Попри велику кількість готових ORM-рішень, таких як Hibernate або EclipseLink, створення власної ORM-системи залишається актуальним завданням для розробників, які прагнуть глибше зрозуміти механізми взаємодії між об'єктно-орієнтованим кодом і реляційними базами даних. Більшість популярних бібліотек є складними для початківців і приховують багато логіки, що ускладнює навчання. До того ж, у невеликих або навчальних проєктах часто немає потреби в повноцінному фреймворку з великою кількістю залежностей. Це створює потребу у простій, зрозумілій та контрольованій ORM-системі, яку можна налаштувати та підтримувати самостійно.

**Мета роботи.** Розробити легку ORM-систему на мові Java, яка забезпечує базову функціональність для збереження, оновлення, видалення та отримання об'єктів із реляційної бази даних без необхідності ручного написання SQL-запитів.

**Результати.** У результаті роботи створено мінімально життєздатну ORM-систему, яка підтримує автоматизовану роботу з базою даних на основі відображення класів та полів. Система реалізує базовий функціонал CRUD-операцій, автоматичне створення таблиць, відображення полів на стовпці за допомогою анотацій, а також механізми для збереження змін. Це дозволяє ефективно управляти сутностями в Java-додатку з мінімальними зусиллями та закладає основу для подальшого розширення функціоналу.

Реалізація системи включає використання механізмів рефлексії [2] для аналізу класів, анотацій для опису структури сутностей, а також SQL-генератора для створення і виконання запитів. Завдяки чітко структурованому коду та розподілу обов'язків між компонентами (EntityManager, SQLBuilder тощо) [3], система є зрозумілою, підтримуваною і легко

доповнюваною. Такий підхід забезпечує не лише працездатність рішення, але й його придатність як навчального прикладу або стартової основи для створення складніших ORM-рішень.

Створена ORM-система може бути інтегрована в невеликі та середні Java-проекти, де використання повнофункціональних бібліотек на кшталт Hibernate є надмірним. Її легковаговість і прозора архітектура роблять її зручною для розробників, які прагнуть повного контролю над обробкою даних. Крім того, система є чудовою базою для експериментів та розширення — наприклад, додавання зв'язків між сутностями, або підтримки альтернативних СУБД. Таким чином, результати проєкту демонструють практичну користь створення власних інструментів, адаптованих під конкретні потреби.

**Висновок.** Таким чином, проведена робота підтверджує як актуальність створення власної ORM-системи для глибшого розуміння принципів роботи з базами даних, так і практичну можливість реалізації такого інструменту одним розробником. Успішне використання мови Java, стандартного JDBC API [4] та механізмів рефлексії дозволило створити мінімально життєздатний продукт, що забезпечує базовий набір функцій для взаємодії з реляційною базою даних. Результати демонструють ефективність обраного підходу та створюють підґрунтя для подальшого розвитку системи — зокрема, додавання розширених можливостей, покращення продуктивності та зручності використання.

### Бібліографія

1. Abba I. What is an ORM – The Meaning of Object Relational Mapping Database Tools. *freeCodeCamp.org*. URL: <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/> (дата звернення: 10.05.2025).
2. GeeksforGeeks. Reflection in Java - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/reflection-in-java/> (дата звернення: 10.05.2025).
3. Use Object-Relational Mapping (ORM) in software development! cleverly in software development!. *Rock the Prototype - Softwareentwicklung & Prototyping*. URL: <https://rock-the-prototype.com/en/software-development/object-relational-mapping-orm/#:~:text=Basic%20concepts:%20The%20core%20principles,data%20manipulation%20and%20better%20maintainability.> (дата звернення: 16.05.2025).
4. GeeksforGeeks. JDBC (Java Database Connectivity) - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/introduction-to-jdbc/> (дата звернення: 10.05.2025).