

ДОСЛІДЖЕННЯ АРХІТЕКТУРНИХ МОДЕЛЕЙ ТА МЕТОДІВ ДЛЯ ПОБУДОВИ ORM ТА ЇХ КОМПАКТНИХ ВАРІАНТІВ НА ПЛАТФОРМІ .NET

Шпорта А. О.

Науковий керівник – к.т.н., доц. Мельнікова Р. В.

Харківський національний університет радіоелектроніки, каф. ПІ

М. Харків, Україна

e-mail: artem.shporta@nure.ua

This research work is aimed at the analysis of architectural models and methods of building compact Object-Relational Mapping (ORM) on the .NET platform. We examine existing approaches to creating compact ORMs, compare their architectural features, and identify key features for developing a common approach to the architecture of such libraries. The main goal of the work is to contribute to the development of more efficient and productive data access tools in the .NET environment.

Область нашої роботи охоплює дослідження та розвиток компактних Object-Relational Mapping (ORM) на платформі .NET. ORM – це методологія, яка дозволяє розробникам працювати з даними бази даних у вигляді об'єктів програмного коду, що спрощує взаємодію з базою даних та полегшує розробку програмного забезпечення.

Дана робота спрямована на аналіз існуючих підходів до побудови компактних ORM, порівняння їхніх архітектурних особливостей та розробку загальних підходів до архітектури таких бібліотек. Ми маємо на меті покращити ефективність та продуктивність розробки програмного забезпечення, що використовує бази даних у .NET-середовищі.

ORM (Object-Relational Mapping) – це технологія програмування, що використовується для зв'язку об'єктно-орієнтованих мов програмування з реляційними базами даних. ORM надає абстракцію над базою даних, що спрощує взаємодію з нею та полегшує розробку програмного забезпечення.

Основні принципи ORM включають відображення класів програмного коду на таблиці бази даних та відображення атрибутів класів на колонки таблиць[1].

ORM також надає можливість виконувати операції з базою даних (створення, читання, оновлення, видалення) з використанням об'єктів програмного коду, що забезпечує більш зрозумілу та підтримувану розробку.

Компактні ORM (Object-Relational Mapping) є важливим компонентом розробки програмного забезпечення, особливо в сучасному світі, де доступ до даних із баз даних є однією з найбільш поширених задач.

Подібні системи ORM створюють можливість розробникам працювати з даними у вигляді об'єктів програмного коду, а не SQL-запитів та рядків даних. Вони спрощують взаємодію з базами даних та дозволяють створювати більш читабельний та підтримуваний код.

Проте, на сьогоднішній день не існує конкретних стандартних інструкцій або загальної архітектурної моделі для розробки компактних ORM. Кожна бібліотека або фреймворк, які надають такий функціонал, може мати свою власну архітектурну концепцію та інтерфейс.

Ця ситуація створює необхідність розробити загальний підхід до архітектури компактних ORM. Дана необхідність зумовлена наступними факторами:

- відсутність загальних стандартів та рекомендацій з архітектури компактних ORM може призвести до великої різноманітності підходів, що ускладнює вибір правильної бібліотеки або фреймворку для проекту;
- розробка загальної архітектурної моделі може допомогти спростити розробку компактних ORM, забезпечуючи консистентність та стандартизацію;
- загальний підхід може полегшити підтримку і розширення існуючих компактних ORM. Розробники можуть бути більш обізнані з архітектурними концепціями та краще розуміти, як працює компонент;
- створення загального підходу до архітектури компактних ORM може сприяти розвитку спільноти та розширенню знань та ресурсів для цього напрямку розробки[2].

Отже, необхідність розробки загального підходу до архітектури компактних ORM визначається потребою в стандартизації, спрощенні розробки, підвищенні продуктивності та покращенні підтримки та розширення таких бібліотек.

Дана робота спрямована на виявлення закономірностей в архітектурах сучасних ORM, на основі закономірностей було сформовано тенденційні патерни для компактних варіантів.

Реалізація наукового дослідження складається з наступних етапів:

- аналіз предметної області;
- аналіз архітектурних особливостей сучасних ORM;
- знаходження закономірностей та логічних висновків що до ідеальної архітектури компактних ORM.

Виведену архітектуру буде повторно проаналізовано та на її основі створено невелику програмну реалізацію.

За результатами роботи було проаналізовано предметну область, та сферу застосування певних ORM, потім була поставлена задача та розроблено стратегію виявлення оптимальних архітектурних патернів для компактних ORM.

Для досягнення цієї мети було розглянуто та проаналізовано архітектури існуючих ORM фреймворків, які широко використовуються в

Java та .NET середовищах. Дослідження охопило як великі, так і компактні ORM системи, включаючи Entity Framework Core для .NET та PetaPoco – як приклад компактного ORM. Після поглибленого аналізу було розроблено приблизну архітектурну схему.

Візуальний SWOT-аналіз перетинів та закономірностей порівняння цих двох ORM зображено на рисунку 1.

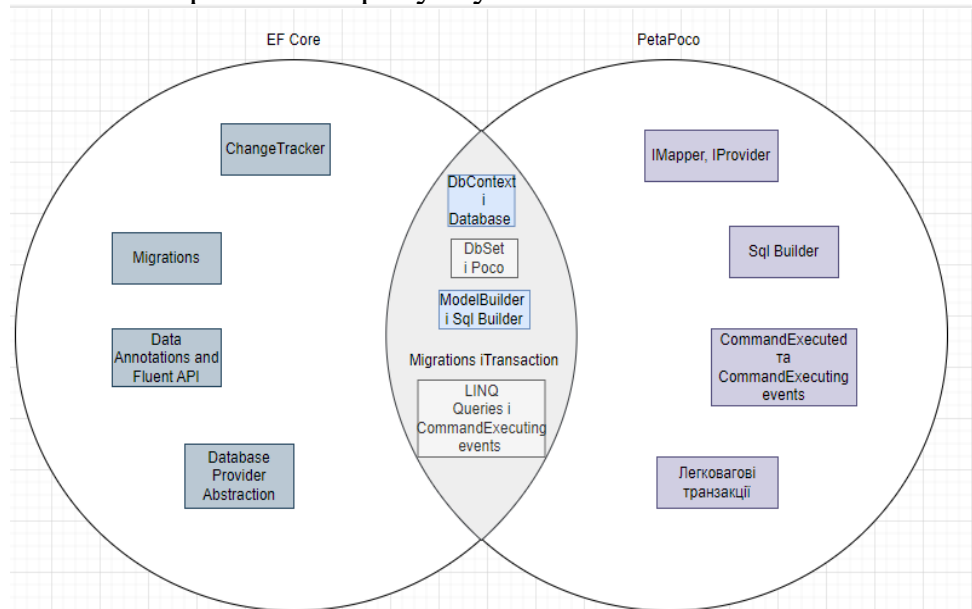


Рисунок 1 – Схема перетину ORM PetaPoco та EF Core

Отже, ці елементи і підходи вирішують завдання, які є центральними для компактних ORM, але можуть бути недостатньо комплексними або занадто простими для повнофункціональних ORM, які вимагають більш розширеного управління даними та відносинами.

Ці структурні патерни разом створюють функціональну основу для компактних ORM, вони роблять їх підходящим для проектів, де необхідний простий у реалізації та ефективний доступ до даних.

На основі аналізу було виявлено певні закономірності які характерні для систем не передбачаючи великих навантажень та довгого часу розробки, надалі ці напрацювання будуть доповнені новими елементами для розробки оптимального архітектурного патерну.

Список використаних джерел:

1. Michał Markiewicz. Repository architecture with OrmLite, 2017. : <https://medium.com/@masztalski/repository-architecture-with-ormlite-fcf7e08ad23e> (дата звернення: 16.05.2023).

2. Mazurova, O., Naboka, A., Shirokopetleva, M. (2021). Research of ACID transaction implementation methods for distributed databases using replication technology. Innovative technologies and scientific solutions for industries, (2 (16)), 19-31. DOI: 10.30837/ITSSI.2021.16.019.