УДК 004.06

Козачок А. О., здобувач вищої освіти;
Гончар В. М., асистент, асистент кафедри інформаційних технологій,
Донецький національний університет імені Василя Стуса

ВИКОРИСТАННЯ ФРЕЙМВОРКУ ENTITY ДЛЯ РОБОТИ З РЕЛЯЦІЙНИМИ БАЗАМИ ДАНИХ У ЗАСТОСУНКАХ НА МОВІ C#

Ключові слова: бази даних, фреймворк Entity, програмування.

Вступ. У сучасному програмуванні реляційні бази даних відіграють важливу роль для того, щоб забезпечити ефективну організацію та зберігання даних. Збільшення обсягу інформації ставить перед розробниками завдання ефективно управляти даними, адаптувати бази даних до різних змін програмного коду та забезпечення високої продуктивності додатків. Саме через це особливого значення набувають технології доступу до даних. Серед цих технологій виділяють потужний інструмент для роботи з реляційними базами даних – Entity Framework. Цей фреймворк забезпечує ефективний і зручний для розробників спосіб доступу до реляційних даних і керування ними, перетворюючи їх на предметно-спеціальні об’єкти.

Актуальність. Фахівці використовують Entity Framework, оскільки він дає змогу взаємодії з базою даних, включає в себе можливості зіставляти класи домену (сутності) зі схемою бази даних, перекладати та виконувати запити LINQ до SQL, відстежувати зміни, що відбулися в об’єктах протягом їх життя, і зберігати відповідні зміни. Entity Framework підтримує роботу з реляційними базами даних, як-от Microsoft SQL Server, MySQL, PostgreSQL, SQL Server Compact, SQLite, Azure Table Storage та IBM Data Server.

Фреймворк був створений для допомоги в роботі з базами даних. Він використовується у мові програмування C#, а саме у надбудові – .Net. Фреймворк дає змогу працювати з базою за допомогою сутностей, а не таблиць. Код з використанням цього фреймворку пишеться набагато швидше і легше. Розробники можуть працювати з даними в об’єктах і властивостями домену, як-от клієнти та адреси клієнтів, не турбуючись про базові таблиці та стовпці, де зберігаються ці дані, завдяки Entity Framework.

Існує три основні підходи для створення фреймворків сутностей:

1. Code First

Цей підхід спочатку націлює на неіснуючу базу даних, а потім створює її. Це дає змогу розробникам визначати та створювати нові моделі за допомогою класів C# та .NET. У цьому підході ви можете використовувати порожню базу даних і додавати таблиці. Під час роботи в режимі Code First концептуальна модель порівнюється зі збереженою моделлю в коді. Entity Framework може отримувати концептуальні моделі на основі типів сутностей і додаткових конфігурацій, які можна вказати. Метадані зіставлення створюються під час виконання з комбінації визначень типу домену та додаткової інформації про

конфігурацію. Entity Framework за потреби створює базу даних на основі метаданих.

2. Model First

Цю модель можна використати для нових проєктів, де немає бази даних. Збереження відбувається за допомогою файла EDMX, він може переглядатися та редагуватися розробником.

3. Database First

Цей підхід є альтернативою для підходу Code First та Model First. Він створює модель і коди у проєкті та з'єднує їх із базою даних і розробником.

Архітектура Entity Framework включає в себе низку ключових компонентів, які спільно працюють для забезпечення ефективного доступу до даних та управління взаємозв'язками між об'єктами та базою даних. Давайте розглянемо ці компоненти більш детально.

Модель даних сутності (EDM):

1. Концептуальна модель.

Описує бізнес-об'єкти та їх зв'язки в термінах класів і властивостей. Відокремлюється від архітектури таблиць бази даних. Являє собою відображення бізнес-логіки та взаємозв'язків між об'єктами.

2. Модель відображення.

Визначає, як концептуальна модель перетворюється на модель зберігання. Об'єднує бізнес-об'єкти та їх зв'язки з таблицями та зв'язками на рівні бази даних.

3. Модель зберігання.

Представляє схему основної бази даних, зокрема таблиці, представлення, ключі та інші об'єкти зберігання.

- **LINQ to Entities:**

Дає змогу писати запити до об'єктів, використовуючи мову LINQ (Language-Integrated Query). Повертає сутності, визначені в концептуальній моделі.

- **Entity SQL:**

Внутрішньо E-SQL перетворює запити на мову запитів SQL, які залежать від сховища даних. Запити E-SQL перетворюються на конкретні мови запитів для системи управління базою даних, як-от T-SQL.

- **Рівень служб об'єктів:**

Включає контекст об'єкта, що представляє сеанс взаємодії між програмою та джерелом даних. Виконує операції, як-от додавання та видалення екземплярів об'єктів і запити до бази даних для збереження зміненого стану.

- **Постачальник даних клієнта сутності:**

Забезпечує абстракцію для доступу до бази даних через об'єктний рівень. Використовує контекст об'єкта для взаємодії з концептуальною моделлю.

- **Постачальник даних ADO.Net:**

Абстрагує інтерфейси ADO.NET для підключення до бази даних на основі концептуальної схеми. Використовує дерево команд для перекладу мов SQL, як-от LINQ, у вирази SQL для виконання в заданій СУБД.

Ці всі компоненти працюють разом, даючи розробникам зручний і потужний інструмент для роботи з реляційними базами даних у .NET-додатках.

Entity Framework використовує файли моделі та відображення для перетворення операцій створення, читання, оновлення та видалення, що виконуються над сутностями та зв'язками в концептуальній моделі, в еквівалентні операції в джерелі даних. Фреймворк навіть підтримує відображення сутностей у концептуальній моделі на збережені процедури у джерелі даних.

Висновки

Отже, у висновку можна сказати, що Entity Framework визначається як потужний інструмент для роботи з реляційними базами даних у сучасному програмуванні, надаючи розробникам зручність і продуктивність у взаємодії з даними. Зокрема, фреймворк дає змогу взаємодіяти з базою даних через сутності, перетворюючи дані на предметно-спеціальні об'єкти.

Розроблений для мови програмування C# та інтегрований у .Net, Entity Framework підтримує три основні підходи: Code First, Model First та Database First. Актуальність використання визначається його ефективністю роботи з різними реляційними базами даних, а також відслідковуванням змін в об'єктах. Використання цього фреймворку прискорює процес написання коду, а також робить роботу набагато ефективнішою.

Entity Framework спрощує взаємодію з базою даних і дає змогу розробникам зосередитися на бізнес-логіці програми, а не працювати з низькорівневими операціями. Він підтримує різних постачальників баз даних, зокрема SQL Server, MySQL, SQLite та ін., що робить його універсальним для різних сценаріїв застосування.

Список використаних джерел

1. Entity Framework documentation hub: вебсайт. URL: https://learn.microsoft.com/en-us/ef/ (дата звернення: 13.11.2023).

2. Фандеєва Я. А., Катєльніков Д. І. Розробка розподіленої інформаційної системи транспортної компанії для оптимізації вантажних перевезень. *Аналіз методів реалізації інформаційної системи:* магістерська робота. ВНТУ. 88 с.

3. Entity Framework VS LINQ to SQL VS ADO NET with stored procedures. URL: https://www.edureka.co/community/204664/entity-framework-vs-linq-sql-ado-net-with-stored-procedures (дата звернення: 13.11.2023).

4. Entity Framework architectural components: вебсайт. URL: https://subscription.packtpub.com/book/programming/9781783550012/1/ch01lvl1sec12/entity-framework-architectural-components