

УДК 017:655.1+004.43/.65

ОСОБЛИВОСТІ РОЗРОБКИ БАЗИ ДАНИХ ПОЛІГРАФІЧНИХ ПІДПРИЄМСТВ НА ОСНОВІ ORM-ТЕХНОЛОГІЙ

Р. І. Петрів, У. П. Пановик, М. О. Майхер, О. І. Сеньків

*Українська академія друкарства
вул. Під Голоском, 19, м. Львів, 79061, Україна*

На основі аналізу можливостей сучасних програмних засобів у роботі запропоновано підходи для створення електронних каталогів поліграфічних підприємств на основі концепцій реляційних баз даних та об'єктно-орієнтованого програмування. Основна увага для цього приділена реалізації ORM (Object-relational mapping, Об'єктно-реляційна проєкція) через застосування об'єктно-орієнтованої технології Entity Framework – структури ORM з відкритим кодом на базі фреймворка .NET.

Як відомо, однією з найбільш популярних систем управління базами даних (СУБД) в світі є Microsoft SQL Server. Дана СУБД застосовується для різних за своїми обсягами проєктів – від невеликих додатків до громіздких проєктів, оскільки характеризується відносною простотою, високою продуктивністю, надійністю і достатнім рівнем безпеки даних. Для безпосередньої розробки бази даних на сервері Microsoft SQL Server пропонується застосування програмного середовища SSMS (SQL Server Management Studio).

Основною перевагою всіх ORM-систем, зокрема Entity Framework, є можливість роботи з сутностями бази даних як з об'єктами мови C#. Суть підходу для створення об'єктно-орієнтованої моделі бази даних полягає в тому, що спочатку створюється її об'єктна модель на мові C#, а потім за допомогою вбудованих функцій фреймворка Entity Framework Core генерується сама база даних на сервері MS SQL Server практично без роботи з самим SQL Server, що на практиці виявляється дуже зручно. Таким чином, забезпечується легка і зручна комунікація з базою даних за допомогою ORM-систем. Модель бази даних має

реляційну структуру і поділена на таблиці, між якими встановлено відповідні зв'язки для забезпечення цілісності даних.

Разом із дослідженням особливостей сучасних технологій управління базами даних для практичної реалізації електронних каталогів поліграфічних підприємств у вигляді системи, яка б поєднувала функціонал такого каталогу та інтерфейс (адмін-панель) для менеджменту технічних, технологічних і людських ресурсів підприємства, окремим завданням є дослідження відповідних технологій веброзробки.

Ключові слова: база даних, ORM-система, Entity Framework, Microsoft SQL Server, електронний каталог.

Постановка проблеми. Використання реляційної бази даних для зберігання об'єктно-орієнтованих даних призводить до семантичного провалу, примушуючи програмістів писати програмне забезпечення, яке повинне вміти як обробляти дані в об'єктно-орієнтованому вигляді, так і вміти зберегти ці дані в реляційній формі. Ця постійна необхідність в перетворенні між двома різними формами даних не тільки значно знижує продуктивність, але й створює труднощі для програмістів, оскільки обидві форми даних накладають обмеження одна на одну. Реляційні бази даних використовують набір таблиць, що представляють прості дані. Додаткова або зв'язана інформація зберігається в інших таблицях. Часто для зберігання одного об'єкта в реляційній базі даних використовують кілька таблиць. Своєю чергою це вимагає застосування операції з'єднання таблиць (JOIN) для отримання всієї інформації, що стосується об'єкта, для її обробки. [1]

Основна задача – необхідно забезпечити роботу з даними в термінах класів, а не таблиць даних і навпаки, перетворити терміни і дані класів в дані, придатні для зберігання в СУБД. Необхідно також забезпечити інтерфейс для CRUD-операцій над даними (створення, читання, оновлення і видалення). Загалом, необхідно позбутися від необхідності писати SQL-код для взаємодії в СУБД.

Аналіз останніх досліджень та публікацій. Рішення проблеми зберігання даних забезпечують реляційні системи управління базами даних. Для створення об'єктно-орієнтованих баз даних зручним

інструментом є застосування технології об'єктно-реляційного відображення (ORM).

ORM (*англ.* Object-Relational Mapping) – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи так звану «віртуальну об'єктну базу даних» [1]. Існують як пропрієтарні, так і вільні реалізації цієї технології.

ORM володіє низкою можливостей та переваг, а саме:

- позбавляє програмістів від написання великої кількості коду, часто одноманітного і схильного до помилок, у зв'язку з чим значно підвищується швидкість розробки. Окрім того, більшість сучасних реалізацій ORM дозволяють програмісту за необхідності самому жорстко задати код SQL-запитів, який буде використовуватися при тих чи інших діях (збереження в базу даних, завантаження, пошук тощо);
- дозволяє програмістам підтримувати узгоджене представлення об'єктів в часі, навіть якщо змінюються джерела, котрі їх надають, приймачі, які їх отримують, і додатки, які до них звертаються;
- керує деталями відображення між набором об'єктів і реляційними базами даних, XML-репозиторіями або іншими джерелами і приймачами даних, одночасно приховуючи від розробників створюваний ними код часто змінюваних деталей пов'язаних інтерфейсів;
- приховує та інкапсулює зміни в самому джерелі даних таким чином, що при зміні джерел даних або їх API-інтерфейсів треба змінювати тільки ORM, а не програми, які використовують ORM. Ця можливість дозволяє розробникам використовувати переваги нових класів за їх появи, а також спрощує розширення додатків на основі ORM. У багатьох випадках зміни ORM можуть включати нові технології і можливості, не потребуючи зміни коду для пов'язаних додатків.

Для забезпечення простоти розробки реляційних баз даних у поєднанні з об'єктно-орієнтованими підходами доречно застосувати для роботи з даними бібліотеку класів Entity Framework – спеціальної об'єктно-орієнтованої технології на базі фреймворка .NET. На відміну від традиційних засобів ADO.NET, які забезпечують створення підключень, команд та інших об'єктів для взаємодії з базами даних, вищий рівень абстракції Entity Framework дозволяє абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища.

Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, то на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами [3].

Центральною концепцією Entity Framework є поняття сутності або entity. Сутність представляє набір даних, асоційованих з певним об'єктом. Тому ця технологія передбачає роботу не з таблицями, а з об'єктами і їх наборами. Будь-яка сутність, як і будь-який об'єкт з реального світу, має низку властивостей. Наприклад, якщо сутність описує людину, то ми можемо виділити такі властивості, як ім'я, прізвище, зріст, вік, вага. Властивості необов'язково представляють прості дані типу int, а й можуть представляти більш комплексні структури даних. І у кожній сутності може бути одна або кілька властивостей, які будуть відрізняти цю сутність від інших і будуть унікально визначати цю сутність. Подібні властивості називають ключами. При цьому сутності можуть бути пов'язані асоціативними зв'язками один-до-багатьох, один-до-одного і багато-до-багатьох, подібно до того, як в реальній базі даних відбувається зв'язок через зовнішні ключі. [3]

Виклад основного матеріалу. Особливістю Entity Framework є використання запитів LINQ (Language Integrated Query) для вибірки даних з БД. За допомогою LINQ можна не тільки отримувати з бази даних певні рядки, що зберігають об'єкти, а й отримувати об'єкти, пов'язані різними асоціативними зв'язками. Іншим ключовим поняттям є Entity Data Model. Ця модель зіставляє класи сутностей з реальними таблицями в базі даних. Entity Data Model складається з трьох рівнів: концептуального, рівня сховища і рівня зіставлення (мапінга). На концептуальному рівні відбувається визначення класів сутностей, які використовуються в програмному додатку. Рівень сховища визначає таблиці, стовпці, відношення між таблицями і типи даних, з якими порівнюється використовувана база даних. Рівень зіставлення є посередником між попередніми двома, визначаючи зіставлення між властивостями класу суті і стовпцями таблиць. Таким чином отримується можливість через класи, визначені у додатку, взаємодіяти з таблицями бази даних [2].

Entity Framework передбачає три можливі способи взаємодії з базою даних:

- Database first: при цьому Entity Framework створює набір класів, які відображають модель конкретної бази даних (рис. 1);
- Model first: спочатку розробник створює модель бази даних, за якою потім Entity Framework створює реальну базу даних на сервері (рис. 2);
- Code first: розробник створює клас моделі даних, які будуть зберігатися в базі даних, а потім Entity Framework за цією моделлю генерує базу даних і її таблиці (рис. 3).

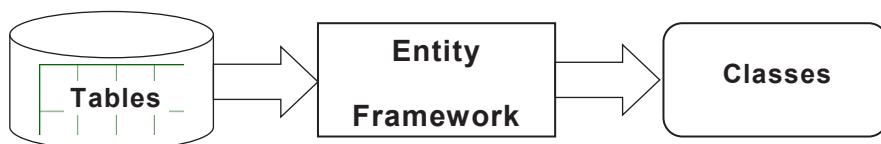


Рисунок 1 Генерування класів доступу до даних для існуючої бази даних

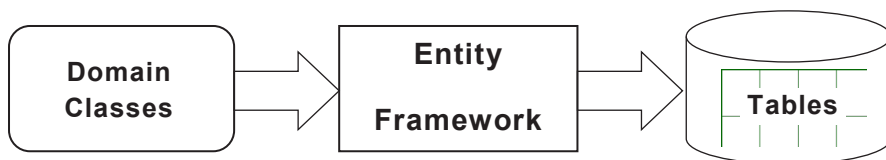


Рисунок 2 Створення бази даних з доменних класів

Розробку бази даних каталогу поліграфічних підприємств доцільно здійснювати на сервері Microsoft SQL Server за допомогою середовища SSMS (SQL Server Management Studio). База даних має реляційну структуру, тобто структурно утворюється таблицями, між якими встановлено відповідні зв'язки для цілісності даних. Зауважимо, що SQL Server є однією з найпопулярніших систем управління базами даних (СУБД) в світі.

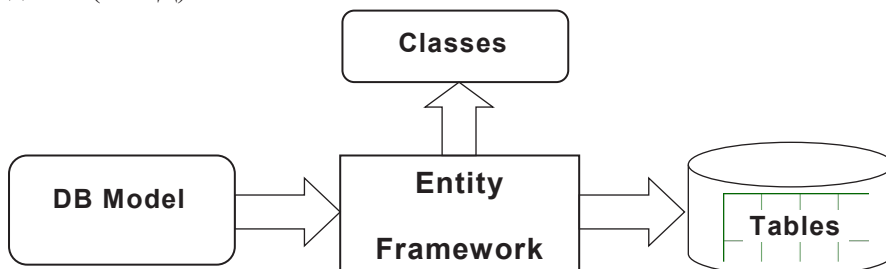


Рисунок 3 Створення бази даних та класів на основі структури моделі бази даних

Дана СУБД підходить для реалізації різних завдань: від невеликих додатків до великих високонавантажених проєктів, оскільки характеризується високою продуктивністю, надійністю, простотою та достатнім рівнем безпеки. Ця система довгий час працювала на платформі Windows, проте, починаючи з версії 16, вона доступна і на Linux.

Виходячи з аналізу структури та функціональних особливостей поліграфічних підприємств, виділимо їх основні сутності як об'єкта для реалізації бази даних, зокрема створення таблиць:

- таблиця «*Підприємства*» – центральна сутність бази даних. Містить інформацію про поліграфічне підприємство;
- таблиця «*Працівники*» – зберігає інформацію про працівника підприємства. Використовується в основному для менеджменту роботи підприємства на панелі адміністрування. Будь-яка інша категорія користувача при пошуку підприємства не матиме доступу до цієї інформації;
- таблиця «*Замовлення*» – зберігає інформацію про замовлення, надіслані для конкретного підприємства від користувачів;
- таблиця «*Поліграфічна продукція*» – в ній перераховані всі можливі види поліграфічної продукції;
- таблиця «*Відгуки про підприємство*» – представляє собою місце для зберігання відгуків користувачів про поліграфічне підприємство на основі досвіду співпраці чи інших факторів. Така функціональність є важливою, оскільки дозволяє новим користувачам легше вибрати підприємство для співпраці.

Зауважимо, що це далеко неповний перелік таблиць, а лише найважливіша їх частина.

При створенні електронних каталогів поліграфічних підприємств особливо актуальними будуть такі переваги ORM:

- наявність явного опису схеми БД, представленого термінами будь-якої мови програмування, що зберігається та редагується в одному місці;
- можливість програмісту оперувати елементами мови програмування (класами, об'єктами, атрибутами, методами), а не елементами реляційної моделі;

- можливість автоматизації створення SQL-запитів, що позбавляє необхідності використання мови опису структури БД і мови маніпулювання даними при проєктуванні БД або при зміні її схеми відповідно;
- відсутня потреба створення нових SQL-запитів при експорті даних в іншу СУБД, оскільки цю процедуру забезпечує драйвер ORM;
- код, який генерується ORM, гіпотетично є перевіреним та оптимізованим, що дозволяє скоротити час на відлагодження і тестування програми;

Для програмної реалізації електронних каталогів пропонується один із найпопулярніших підходів – Code First. Суть підходу полягає в тому, що спочатку створюється не сама база даних, а її об'єктна модель на мові C#, а потім за допомогою вбудованих функцій фреймворка генерується власне база даних на сервері MS SQL Server.

Висновки. Проведення певних маніпуляцій з доменною моделлю на мові C# при використанні підходу Entity Framework Code First дозволяє згенерувати базу даних практично без роботи з самим MS SQL Server, що на практиці виявляється дуже зручно.

Основна ідея всіх ORM-систем, таких як Entity Framework, полягає не просто у створенні доменних моделей аплікації і генеруванні з них бази даних. Основна перевага – це можливість роботи з сутностями бази даних як з об'єктами мови C#. Тому недостатньо мати тільки набір моделей, потрібно мати централізований механізм для комунікації з ними. Таким механізмом у Entity Framework є контекст бази даних – клас, який унаслідкується від визначеного розробником базового класу. У визначеному контексті (класі) бази даних є колекції всіх моделей, які виступають свого роду об'єктно-орієнтованими представленнями таблиць бази даних. Оперуючи попередньо створеними колекціями моделей бази даних, ми певним чином оперуємо таблицями, якщо додати нових запис у колекцію і зберегти зміни, то запис додасть у базу даних. Саме ця рефлексивність і є основною перевагою ORM-систем.

Недоліком ORM вважають низьку продуктивність, пов'язану із збільшенням обсягу програмного коду та зниженню швидкодії програми. Справді, це має місце, однак слід врахувати, що більшість OR спрямовані на опрацювання значної кількості різноманітних сценаріїв опрацювання даних, ніж у випадку використання одного програмного додатку.

Список використаних джерел

1. Об'єктно-реляційне відображення. URL: <https://uk.wikipedia.org/wiki>.
2. Огляд Entity Framework. URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/ef/overview>.
3. Тимофєєва А.Є. Дослідження Entity Framework – технології платформи .NET: матеріали міжнародної науково-практичної конференції «Перспективні напрямки наукової думки». Том 5 (м. Тернопіль, 18 квітня 2018 року). Тернопіль, 2018. – С.52-53.

REFERENCES

1. Obiektno-reliatsiine vidobrazhennia. URL: <https://uk.wikipedia.org/wiki>. (in Ukrainian)
2. Ohliad Entity Framework. URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/ef/overview>. (in Ukrainian)
3. Tymofieieva A. Ye. (2018). Doslidzhennia Entity Framework – tekhnolohii platformy .NET: materialy mizhnarodnoi naukovo-praktychnoi konferentsii «Perspektyvni napriamky naukovoï dumky». Tom 5 (m. Ternopil, 18 kvitnia 2018 roku). Ternopil – S.52-53. (in Ukrainian)

DOI 10.32403/2411-9210-2023-2-50-32-40

**FEATURES OF DATABASE DEVELOPMENT
OF PRINTING COMPANIES
BASED ON ORM TECHNOLOGIES**

R.I. Petriv, U.P. Panovyk, M.O. Maikher, O.I. Senkiv
*Ukrainian Academy of Printing,
19, Pid Holoskom St., Lviv, 79020, Ukraine
r_petriv@ukr.net*

Based on the analysis of the capabilities of modern software tools, the paper proposes approaches for creating electronic catalogues of printing companies based on the concepts of relational databases and object-oriented programming. The main attention to do this is paid to the implementation of ORM (Object-Relational Mapping, Object-Relational Projection) through the

use of Entity Framework object-oriented technology – an open source ORM structure based on the .NET framework.

As it is known, one of the most popular database management systems (DBMS) in the world is Microsoft SQL Server. This DBMS is used for projects of various sizes – from small applications to bulky projects, as it is characterized by relative simplicity, high performance, reliability and a sufficient level of data security. For direct development of the database on the Microsoft SQL Server, the use of the SSMS (SQL Server Management Studio) software environment is suggested.

The main advantage of all ORM systems, in particular the Entity Framework, is the ability to work with database entities as C# language objects. The essence of the approach to creating an object-oriented database model is that its object model is first created in the C# language, and then with the help of the built-in functions of the Entity Framework Core framework, the database itself is generated on the MS SQL Server almost without working with SQL Server itself, which turns out to be very convenient in practice. Thus, easy and convenient communication with the database using ORM systems is provided. The database model has a relational structure and is divided into tables, between which appropriate relationships are established to ensure data integrity.

Together with the study of the features of modern database management technologies for the practical implementation of electronic catalogues of printing companies in the form of a system that would combine the functionality of such a catalogue and an interface (admin-panel) for the management of technical, technological and human resources of the company, a separate task is the study of relevant web development technologies.

Keywords: *database, ORM system, Entity Framework, Microsoft SQL Server, electronic catalogue.*

Стаття надійшла до редакції 16.05.2024.

Received 16.05.2024.