DOI: 10.58168/MoInSyTe2024_37-43
УДК 004.9

КОНСТРУИРОВАНИЕ БАЗЫ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ENTITY FRAMEWORK ПОДХОД CODE FIRST

И.А. Бунеев[1], Н.Ю. Юдина[1]

[1]ФГБОУ ВО «Воронежский государственный лесотехнический университет имени Г.Ф. Морозова»

Аннотация. В статье рассматривается применение фреймворк ORM Entity Framework. Entity Framework избавляет разработчиков от написания большого кода доступа к данным. При использовании данного фреймворка создается слой абстракции, так называемая EDM-модель, определяющая правила отображения объектов на базу данных. Code First используется, когда база данных еще не создавалась. На рисунке 1 представлен подход Code First к созданию базы данных на основе класса контекста и классов, описывающих сущности, т.е. сначала необходимо создать приложение на C#, а затем Entity Framework из имеющегося кода создаст базу данных

Ключевые слова: Entity Framework, Code First, база данных, класс, контекст.

CONSTRUCTION OF A DATABASE USING ENTITY FRAMEWORK CODE FIRST APPROACH

I.A. Buneev[1], N.Yu. Yudina[1]

[1]Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. This article discusses the use of the ORM Entity Framework. Entity Framework saves developers from writing a lot of data access code. When using this framework, an abstraction layer is created, the so-called EDM model, which defines the rules for mapping objects to the database. Code First is used when the database has not yet been created. Figure 1 shows Code First's approach to creating a database, which is created based on domain classes and a context class, i.e. First you need to create an application in C#, and then Entity Framework will create a database from the existing code.

Keywords: Entity Framework, Code First, database, class, context.

Entity Framework представляет собой объектно-ориентированную технологию от компании Microsoft для доступа к данным. Данная технология является ORM-инструментом (object-relational mapping - отображения данных на реальные объекты) и позволяет работать с базами данных, представляя собой более высокий уровень абстракции.

Центральной концепцией Entity Framework является понятие сущности. Сущность – некоторый набор данных, связанный с объектом.

Технология ускоряет разработку и реализует паттерн «Репозиторий», что позволяет создавать приложения, взаимодействующие с реляционными СУБД с помощью строго типизированных объектов .NET

Подход Code First — это подход к разработке программного обеспечения, в котором сначала создается код реализации системы, а затем на его основе автоматически генерируется соответствующая схема базы данных или контракт между компонентами. Entity Framework имеет возможность автоматически генерировать таблицы в базе данных по указанным в контексте подключения классам.

На рис. 1 представлен подход Code First к созданию базы данных на основе класса контекста и классов, описывающих сущности, т.е. сначала необходимо создать приложение на C#, а затем Entity Framework из имеющегося кода создаст базу данных.

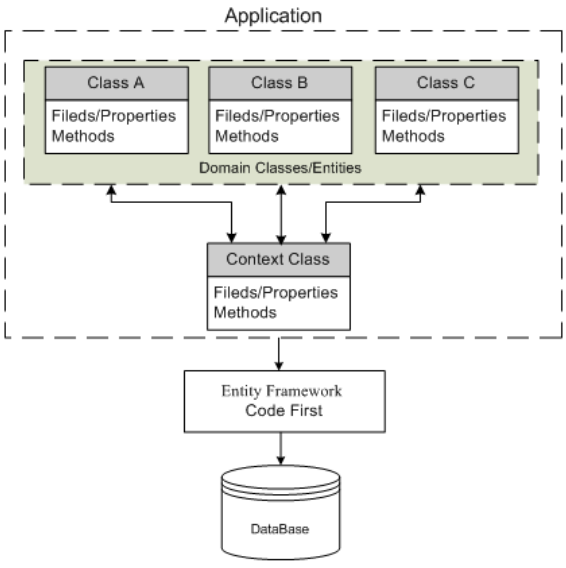

Рисунок 1 – Схема работы подхода Code First в Entity Framework

Библиотеку Entity Framework можно установить, используя диспетчер пакетов NuGet.

Рисунок 2 – Entity Framework

Перед началом работы необходимо создать класс – контекст подключения к базе данных.

```
public class DiplomDatabase: DbContext
    {
        public DiplomDatabase (): base("server = (localdb)\\MSSQLLocalDB;
Database=Diplom;Integrated Security=True;")
        {
        public DbSet<Material> Materials { get; set; }
        public DbSet<User> Users { get; set; }
        public DbSet<Calc> Calcs { get; set; }
        public DbSet<Resistor> Resistors { get; set; }
        public DbSet<Condensator> Condensators { get; set; }
    }
}
```

Рисунок 3 – Класс контекста базы данных DiplomDatabase.cs

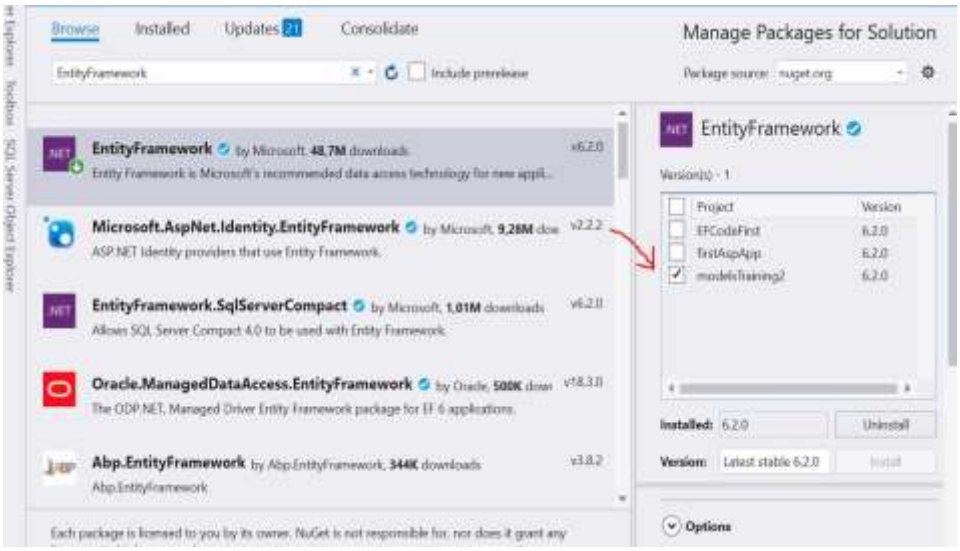

Рисунок 4 – Подключение библиотеки Entity Framework в проект

При первом обращении к контексту базы данных выполняется проверка на то, что указанная в строке подключения БД существует. Если она отсутствует, то происходит создание новой и добавление исходных данных (рисунок 5).

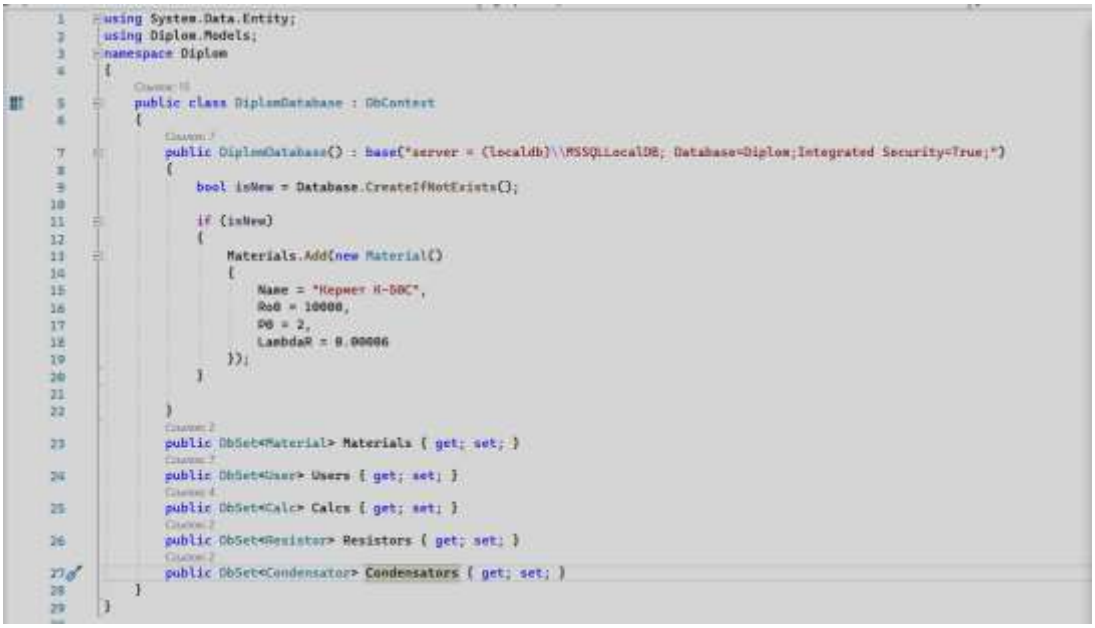

Рисунок 5 – Контекст базы данных с проверкой на существование в конструкторе класса

DiplomDatabase является наследником класса DbContext, который используется для взаимодействия с базой данных.

Каждый таблица описывается типом DbSet с указанием соответствующего ей класса. Entity Framework при создании базы данных анализирует структуру классов-сущностей, которые используются в программе, и создает соответствующие таблицы.

Например, в нашем случае класс User выглядит так (рисунок 6).

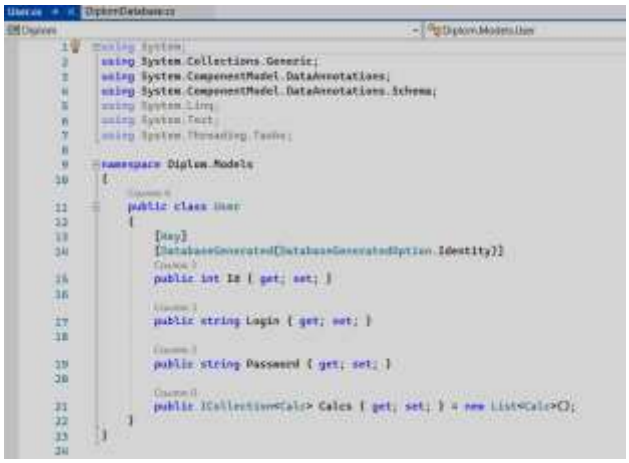

Рисунок 6 – Класс User

Атрибуты «Key» и «DatabaseGenerated» указывают на то, что данное свойство является первичным ключом с автоинкрементом.

Класс «Users» содержит свойства «Id», «Login» и «Password». Соответственно в таблице, которая будет создана при запуске программы, будет три

столбца (рисунок 7).

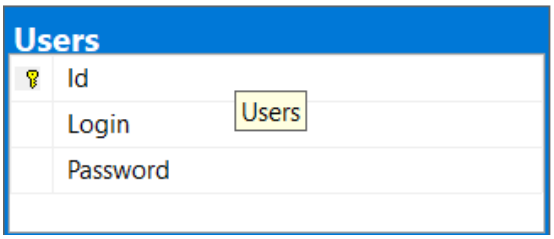

Рисунок 7 – определение первичного ключа

Свойство Calcs позволяет установить отношение один ко многим с таблицей Calcs. Данные для справочного материала добавляются сразу при создании БД. Остальные таблицы заполняет пользователь во время работы с программой. Пример заполнения таблицы приведен на рисунке 8.

```
public DiplomDatabase() : base("server = (localdb)\\MSSQLLocalDB; Database=Diplom;Integrated Security=True;
{
    bool isNew = Database.CreateIfNotExists();

    if (isNew)
    {
        Materials.Add(new Material()
        {
            Name = "Кермет К-50С",
            Ro0 = 10000,
            P0 = 2,
            LambdaR = 0.00006
        });
    }
```

Рисунок 8 – Описание таблицы Materials

Класс Condensator описывает таблицу (рисунок 9), содержащую характеристики конденсаторов, которые могут быть использованы при построении схемы.

```
public class Condensator
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }
        public int CalcId { get; set; }
        public Calc Calc { get; set; }

        public double C { get; set; }
        public double deltaC { get; set; }

        public double d { get; set; }
        public double C0u { get; set; }
        public double deltaCt { get; set; }
        public double deltaS { get; set; }
        public double C0tech { get; set; }

        public double d1 { get; set; }
        public double S { get; set; }
        public double K { get; set; }

        public double Sb { get; set; }
        public double Lb { get; set; }
        public double Ln { get; set; }
        public double Ld { get; set; }

        public double Sitog { get; set; }

    }
}
```

Рисунок 9 – Класс Condensator

Описание используемого материала проводников и контактных площадок представлено в классе Material.cs (рисунок 10).

```
public class Material
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    Ссылок: 0
    public int Id { get; set; }
    Ссылок: 2
    public string Name { get; set; }
    Ссылок: 2
    public double Ro0 { get; set; }
    Ссылок: 2
    public double P0 { get; set; }
    Ссылок: 2
    public double LambdaR { get; set; }
}
}
```

Рисунок 10 – Класс Material.cs

Таблица базы данных, содержащая сведения о резисторах описана в классе Resistor.cs (рисунок 11).

```
public class Resistor
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }
    public int CalcId { get; set; }
    public Calc Calc { get; set; }
    public double R { get; set; }
    public double deltaR { get; set; }
    public double P { get; set; }
    Ссылок: 1
    public double Kf { get; set; }
    Ссылок: 4
    public double Bp { get; set; }
    Ссылок: 2
    public double BToch { get; set; }
    Ссылок: 2
    public double LRasch {get; set; }
    Ссылок: 3
    public double L { get; set; }
    Ссылок: 2
    public double S { get; set; }
}
}
```

Рисунок 11 – Класс Resistor.cs

Entity Framework Code First создаст базу данных, используя параметр заданный в классе DiplomDatabase.cs и переданный в конструктор класса DBContext. База данных создается в SQL Express, т.е. в локальной базе данных. Он автоматически создаст четыре таблицы Resistor, Material, Condensator, User. Для просмотра базы данных откройте «Обозреватель объектов SQL Server» и выберите свою базу.

Список литературы

1. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. - СПб.: Питер, 2019. - 896 с.

2. Бедердинова, О. И. Программирование на языках высокого уровня : учеб. пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. – Москва : ИНФРА-М, 2019.– 159 с. - Текст: электронный. - URL: https://znanium.com/catalog/product/1044396. – Режим доступа: по подписке.

3. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. – Москва: ФОРУМ: ИНФРА-М, 2020. – 200 с. – (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1069921 – Режим доступа: по подписке.

4. Lerman, J., Miller, R. Programming Entity Framework: Code First/ J. Lerman, R. Miller; - O'Reilly Media, Inc, 2011 – 177 p.- ISBN: 978-1449312947

5. Полуэктов А.В., Макаренко Ф.В., Ягодкин А.С. Использование сторонних библиотек при написании программ для обработки статистических данных // Моделирование систем и процессов. – 2022. – Т. 15, № 2. – С. 33-41.

References

1. Richter, J. CLR via C#. Programming on the Microsoft .NET Framework 4.5 in C# / J. Richter. - St. Petersburg: Peter, 2019. - 896 p.

2. Bederdinova, O. I. Programming in high-level languages: textbook. allowance / O.I. Bederdinova, T.A. Mineeva, Yu.A. Vodovozova. – Moscow: INFRA-M, 2019. – 159 p. - Text: electronic. - URL: https://znanium.com/catalog/product/1044396. – Access mode: by subscription.

3. Khorev, P.B. Object-oriented programming with examples in C#: textbook / P.B. Khorev. – Moscow: FORUM: INFRA-M, 2020. – 200 p. – (Higher education: Bachelor's degree). - ISBN 978-5-00091-680-3. - Text: electronic. - URL: https://znanium.com/catalog/product/1069921 – Access mode: by subscription.

4. Lerman, J., Miller, R. Programming Entity Framework: Code First/ J. Lerman, R. Miller; - O'Reilly Media, Inc, 2011 – 177 p.- ISBN: 978-1449312947

5. Poluektov A.V., Makarenko F.V., Yagodkin A.S. The use of third-party libraries when writing programs for processing statistical data // Modeling of systems and processes. - 2022. – Vol. 15, No. 2. – pp. 33-41.