**Лабораторная работа №4: Основы проектирования приложений Windows Forms**

В рамках данной лабораторной работы необходимо создать базу данных для будущего приложения при помощи модуля Entity Framework Core. База данных создавалась на тему «Зоомагазин».

В первую очередь модуль необходимо установить. Для этого необходимо найти его в менеджере пакетов NuGet:

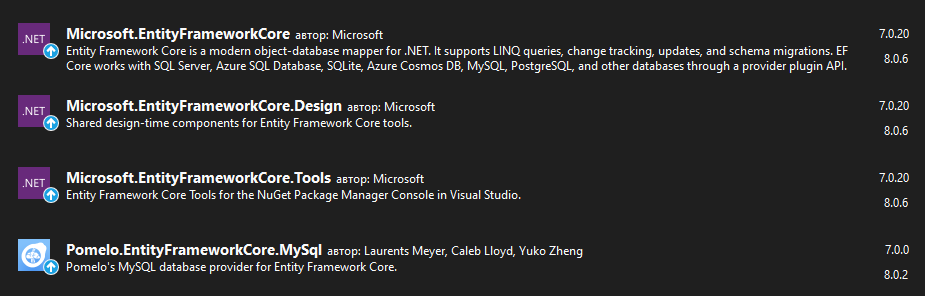


Рисунок 1. Набор модулей, необходимых для разработки приложения

Также для создания миграции базы данных с последующим переносом её в MySQL необходимо установить расширение Tools для EFC и MySQL коннектор от Pomelo.

Следующим этапом разработки БД стало создание классов для таблиц. Была определена следующая структура базы данных:

1. Categories – таблица для хранения категорий товаров в зоомагазине.
2. Customers – таблица для хранения информации о покупателях.
3. Orders – таблица заказов покупателей.
4. Order\_Items – таблица позиций заказов.
5. Products – таблица товаров в зоомагазине.
6. Suppliers – таблица с информацией о поставщиках товаров.

Таблица 1 – Содержимое таблицы Categories.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| name | text | Название |
| description | text | Описание |

Был создан следующий класс для данной таблицы:

public class Categories

{

[Key]

public int id { get; set; }

[Required]

public string name { get; set; }

public string description { get; set; }

public List<Products> products { get; set; }

}

Таблица 2 – Содержимое таблицы Customers.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| first\_name | text | Имя клиента |
| last\_name | text | Фамилия клиента |
| email | text | Почта клиента |
| phone | text | Мобильный номер |
| address | text | Адрес проживания |

Был создан следующий класс для данной таблицы:

public class Customers

{

[Key]

public int id { get; set; }

[Required]

public string first\_name { get; set; }

public string last\_name { get; set; }

public string email { get; set; }

public string phone { get; set; }

public string address { get; set; }

public List<Orders> orders { get; set; }

}

Таблица 3 – Содержимое таблицы Orders.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| order\_date | text | Дата заказа |
| total | int | Стоимость |
| status | int | Статус заказа |
| customer\_id | int | Номер клиента |

Был создан следующий класс для данной таблицы:

public class Orders

{

[Key]

public int id { get; set; }

[Required]

public string order\_date { get; set; }

public int total { get; set; }

public int status { get; set; }

public Customers customer\_ { get; set; }

public List<Order\_Items> order\_items { get; set; }

}

Таблица 4 – Содержимое таблицы Order\_Items.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| quantity | int | Количество товаров |
| total | int | Стоимость |
| orders\_id | int | Номер заказа |
| products\_id | int | Номер товара |

Был создан следующий класс для описанной таблицы:

public class Order\_Items

{

[Key]

public int id { get; set; }

[Required]

public int quantity { get; set; }

public int total { get; set; }

public Orders orders\_ { get; set; }

public Products products\_ { get; set; }

}

Таблица 5 – Содержимое таблицы Products.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| name | text | Название товара |
| description | text | Описание товара |
| price | int | Цена товара |
| quantity | int | Количество |
| categories\_id | int | Номер категории |
| suppliers\_id | int | Номер поставщика |

Был создан следующий класс:

public class Products

{

[Key]

public int id { get; set; }

[Required]

public string name { get; set; }

public string description { get; set; }

public int price { get; set; }

public int quantity { get; set; }

public Categories categories\_ { get; set; }

public Suppliers suppliers\_ { get; set; }

public List<Order\_Items> order\_items { get; set; }

}

Таблица 6 – Содержимое таблицы Suppliers.

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| id | int | Уникальный номер |
| name | text | Имя/название заказчика |
| email | text | Почта заказчика |
| phone | text | Номер телефона |
| address | text | Адрес заказчика |

Для описанной таблицы создан следующий класс:

public class Suppliers

{

[Key]

public int id { get; set; }

[Required]

public string name { get; set; }

public string email { get; set; }

public string phone { get; set; }

public string address { get; set; }

public List<Suppliers> suppliers { get; set; }

}

Следующим этапом создания базы данных стало создание контекстного менеджера (DataContext.cs), который подключится к MySQL, сделает миграцию созданных классы и сохранит новую БД в виде таблиц в MySQL. В контекстном менеджере представлены все ранее созданные классы в следующем формате (который соответствует шаблонам Entity Framework Core):

public DbSet<Categories> Categories { get; set; }

public DbSet<Customers> Customers { get; set; }

public DbSet<Orders> Orders { get; set; }

public DbSet<Order\_Items> Order\_Items { get; set; }

public DbSet<Products> Products { get; set; }

public DbSet<Suppliers> Suppliers { get; set; }

Также для подключения использовался коннектор MySQL, который называется OnConfiguring, в котором необходимо указать данные для подключения к MySQL:

protected override void OnConfiguring(DbContextOptionsBuilder optionBuilder)

{

optionBuilder.UseMySql ("server=localhost;user=root;password=root;database=zooshop",

new MySqlServerVersion(new Version(10, 3, 0)));

}

Последний этап создания БД – формирование миграции базы данных и сохранение полученного результата в MySQL. Для этого необходимо в консоли создать миграцию с помощью команды Add-Migration и обновить MySQL с помощью команды Update-Database:

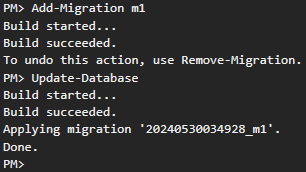


Рисунок 2. Создание миграции базы данных

Таким образом получилось создать базу данных MySQL, которая будет использована при разработке полноценного приложения.

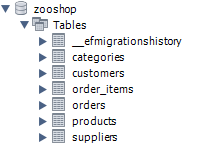


Рисунок 3. Результат создания БД в MySQL