**Лабораторная работа №5: Разработка приложения в Visual Studio с использованием MySQL**

В рамках данной лабораторной работы необходимо разработать приложение для работы с базой данных, которая была создана в предыдущей работе. Приложение должно состоять из главной формы и форм для отображения и редактирования таблиц. Таким образом предполагается наличие 6 форм (для формы заказов и содержимого заказа используется 1 общая форма).

Разработка приложения началась с создания стартовой формы. На данной форме представлены кнопки для перемещения по другим формам, а также есть кнопка для закрытия программы (см. рис. 1).

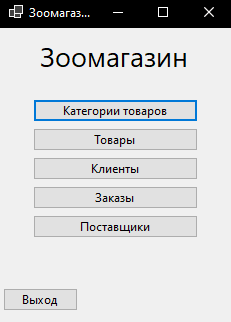


Рисунок 1. Главная форма приложения

Далее была создана форма для таблицы категорий товаров (см. рис. 2). В форме присутствуют кнопки добавления, обновления и удаления данных. Для добавления данных необходимо заполнить поля, которые представлены выше таблицы. Таблица выводится с помощью элемента dataGridView, который является стандартным элементом Visual Studio. Для обновления достаточно изменить содержимое элемента dataGridView и нажать кнопку обновления. Для удаления необходимо выбрать нужную строку и нажать кнопку удаления.

Код всех функций основан на Entity Framework Core. Например, для создания массива данных из таблицы используется следующий код:

var clist = db.Categories.ToList();

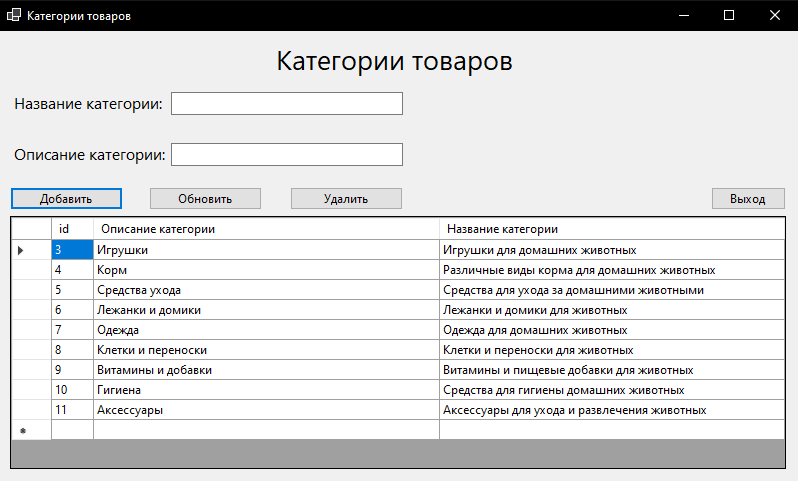
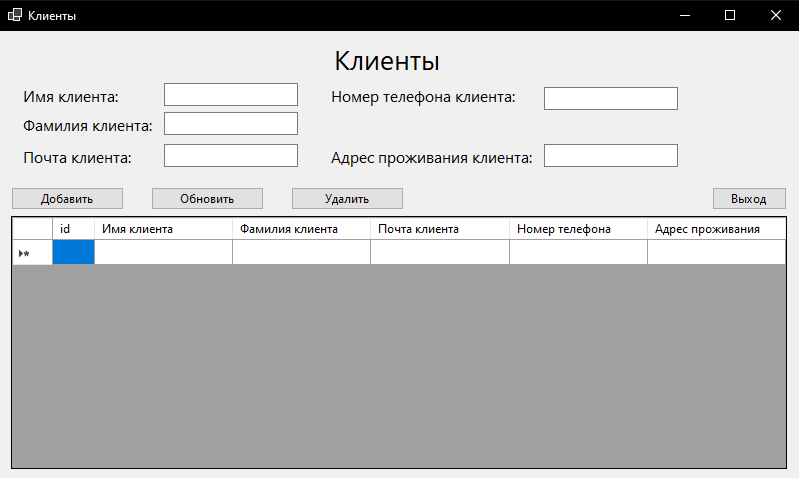


Рисунок 2. Форма категорий товаров

Формы для редактирования таблиц клиентов и поставщиков сделаны по тому же принципу (см. рис. 3).



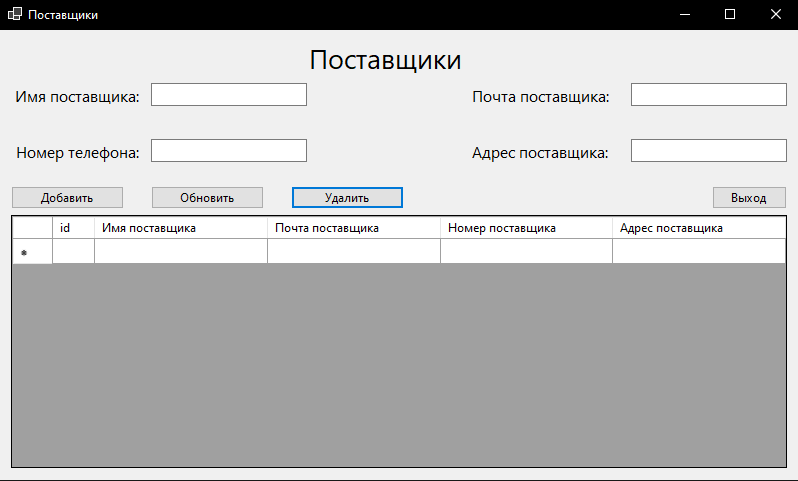


Рисунок 3. Формы для редактирования клиентов и поставщиков

Для товаров необходимо также сделать возможность выбора категорий из выпадающего списка. Для этого нужен элемент comboBox, в который будут вноситься данные из таблицы Categories. В таблице отображаются номера категорий, а не названия (см. рис. 4).

Также в этой таблице есть связь с другой таблицей – с категориями. Для добавления связи в запись сначала необходимо получить ссылку на строку в таблице Categories, а затем её добавить в новый объект класса Products. Тоже самое касается и связи товара с поставщиками (то есть с таблицей Suppliers).

Для отображения данных из связанных таблиц необходимо добавить свойство Include в объявление массива данных для таблицы:

var plist = db.Products.Include(c => c.categories\_).Include(s => s.suppliers\_).ToList();

Изменять значения для категории и поставщиков после добавления нельзя из-за особенностей работы Entity Framework Core.

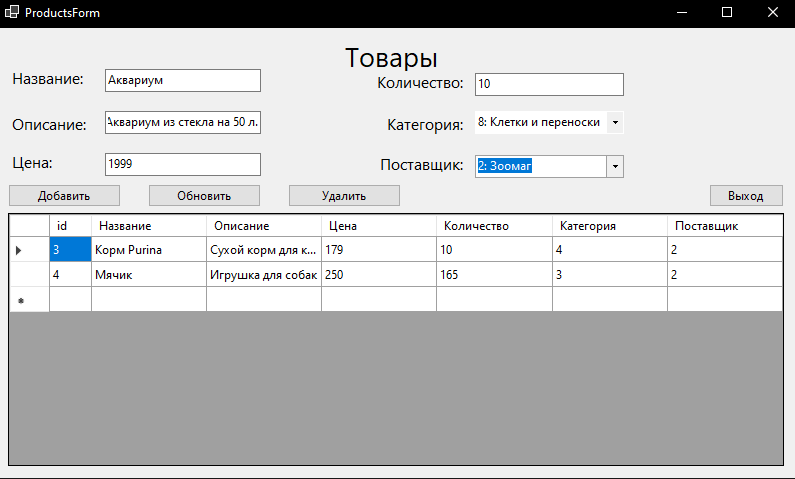


Рисунок 4. Форма товаров в приложении

Форма для работы с заказами работает аналогично вышеописанным (см. рис. 5).

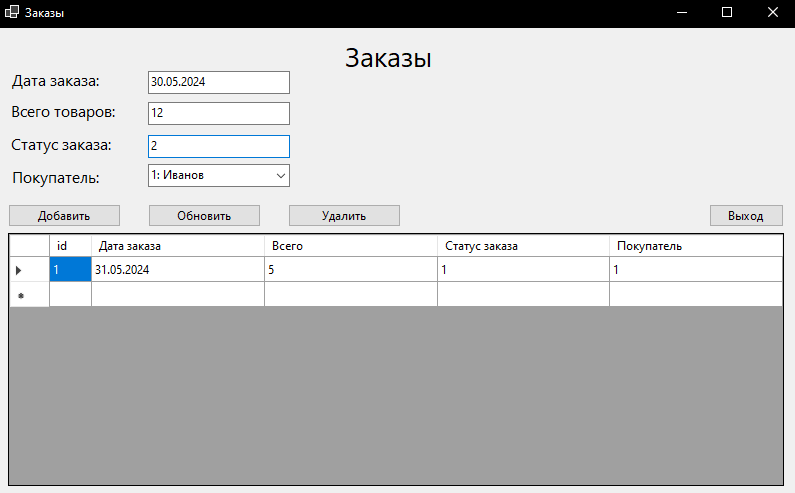


Рисунок 5. Форма работы с заказами