

Лабораторная работа №12

Доступ к базе данных с помощью LINQ to SQL в проекте Visual C#

Вначале создадим базу данных городов, содержащую два поля: название города и численность его населения. Затем организуем доступ к этой базе данных с помощью **LINQ to SQL** и создадим запрос на извлечение коллекции городов, численность населения в которых превышает 500 000 человек. Эту задачу можно было бы решить, организовав LINQ-запрос через набор данных **DataSet**, как мы это делали в предыдущей работе, однако имеется и другой подход.

Вначале запустим **Visual Studio** и выберем проект шаблона **Windows Forms Application**, укажем любое имя. Далее, попав в конструктор формы, из панели элементов **Toolbox** перетащим элемент управления для отображения и редактирования табличных данных **DataGridView**, на этот элемент в конечном итоге будет попадать результат запроса.

Теперь создадим базу данных SQL Server. Для этого в меню **Project** (Проект) выберем команду **Add New Item** (Добавить новый элемент). В появившемся окне выберем элемент **База данных, основанная на службах**, а в поле **Name** укажем имя базы данных Города.mdf. Далее в окне мастера настройки источника данных зададим тип модели базы данных — **Набор данных**. Затем согласимся на сохранение строки подключения в файле конфигурации приложения. Теперь после щелчка на кнопке **Готово** будет создан пустой набор данных. Этот набор данных Города.mdf теперь будет виден в окне **Solution Explorer** (Обозреватель решений).

Чтобы заполнить этот набор данных, дважды щелкнем мышью по значку **Города.mdf**; таким образом, мы попадаем в окно **Server Explorer/Database Explorer** (Обозреватель серверов/Обозреватель баз

данных). Здесь в контекстном меню узла **Таблицы** выберем команду **Добавить новую таблицу**. В результате мы попадаем уже в другое окно — **dbo.Table1**, где зададим имена двух столбцов: **Город** и **Население**.

При сохранении (<Ctrl>+<S>) пустой таблицы появится запрос на выбор имени для таблицы, здесь мы зададим имя **Города**.

Теперь будем заполнять сформированную таблицу. Для этого в **Обозревателе серверов** щелкнем правой кнопкой мыши на узле **Города** (имя нашей таблицы) и в появившемся контекстном меню выберем команду **Показать таблицу данных**. Теперь в окне **Города** мы имеем возможность заполнять нашу таблицу.

На этом этапе задача создания базы данных и заполнения в ней таблицы городов выполнена. Приступаем к организации запроса к таблице городов. Как уже указывалось ранее, LINQ-запрос можно построить через набор данных **DataSet**, а можно LINQ-запрос организовать с помощью классов LINQ to SQL. Эти классы сопоставляются с таблицами и представлениями базы данных и называются *классами сущностей* **DataContext**. Класс сущности сопоставляется с записью, а отдельные свойства класса сущности сопоставляются с отдельными столбцами, образующими запись. Сказанное, вероятно, звучит запутанно, но практически сводится к перетаскиванию мышью созданной нами таблицы Города из окна **Server Explorer/Database Explorer** (Обозреватель серверов/Обозреватель баз данных) на так называемый Object Relational Designer (реляционный конструктор объектов). В результате получим класс сущностей именно для нашей таблицы Города, наследованный от базового класса **DataContext**, и в тексте нашей программы уже легко сможем строить LINQ-запросы, обращаясь к объекту класса сущностей.

Чтобы получить в нашем проекте реляционный конструктор объектов, в меню **Project** выберем команду **Add New Item** (Добавить новый элемент), а в появившемся одноименном окне — шаблон (элемент)

LINQ to SQL Classes. В поле **Name** укажем имя файла Сущности.dbml и щелкнем на кнопке **Add**.

Теперь, как мы уже говорили, просто перетаскиваем мышью таблицу Города из окна **Server Explorer/Database Explorer** (Обозреватель серверов/Обозреватель баз данных) на реляционный конструктор объектов. Реляционный конструктор объектов создает классы и применяет специфические для LINQ to SQL атрибуты, чтобы иметь функциональные возможности LINQ to SQL (возможности передачи данных и редактирования).

Пропишем программный код, следуя шагам:

1. С помощью объекта базового класса **DataContext** будем иметь доступ к таблице базы данных

```
private СущностиDataContext БД = new СущностиDataContext();
```

2. Получим коллекцию записей из таблицы базы данных

```
var a = from gor in БД.Города
        where Convert.ToInt32(gor.Население) > 500000
        select gor;
```

ИЛИ `select new { gor.Название, gor1.Население }` если необходимо выбрать не все поля, а только необходимые.

3. Присвоим источник данных для сетки:

```
dataGridView1.DataSource = a;
```

Задание для самостоятельного выполнения

1. Создать таблицу «Сотрудники» и заполнить ее данными:

№ П/П	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ДАТА РОЖДЕНИЯ	ОБРАЗОВ	ДАТА ЗАЧИСЛЕНИЯ	ДОЛЖНОСТЬ	СПЕЦИАЛЬНОСТЬ
72	Уласевич	Мария	Ивановна	05.01.1955	среднее спец.	01.08.1973	бухгалтер	бухгалтер
73	Юркевич	Любовь	Васильевна	14.07.1957	среднее спец.	25.06.1984	контролер	гидротехник
11	Будыкина	Галина	Константиновна	12.05.1961	среднее спец.	28.04.1990	контролер	техник-электрик
13	Вакарев	Вячеслав	Васильевич	23.03.1941	среднее спец.	16.02.1992	старший контролер	техник радиосвязи
64	Трусов	Игорь	Евгеньевич	12.08.1948	среднее спец.	28.06.1996	контролер	электрик
18	Гринько	Наталья	Никитична	01.01.1960	среднее спец.	24.04.1998	контролер	техник-плановик
5	Бибко	Владимир	Михайлович	07.07.1952	среднее спец.	10.06.1959	техник	техник-строитель
10	Бородько	Татьяна	Николаевна	02.06.1969	среднее спец.	01.06.1986	контролер	инженер-экономист

2. Сделайте запросы для выборки данных
 - 2.1. Выберите сотрудников, у которых юбилей в этом году
 - 2.2. Выберите сотрудников, рожденных в 3 квартале года
 - 2.3. Добавьте поле «Оклад» и определите максимальный и минимальный оклады для каждой должности
 - 2.4. Добавьте поле «Отдел», сгруппируйте сотрудников по отделу и определите Фамилии сотрудников с минимальным и максимальным окладом по каждому отделу