

ISSN 2306-1561

Automation and Control in Technical Systems (ACTS)

2014, No 2, pp. 41-59.

DOI: 10.12731/2306-1561-2014-2-5



Display on the screen data base tables

Chernov Edgar Aleksandrovich

Russian Federation, Ph. D., Associate Professor, Department of «Automated Control Systems».

Moscow Automobile & Road construction State Technical University, 125319, Russian Federation, Moscow, Leningradsky prospekt, 64. Tel.: +7 (499) 151-64-12. <http://www.madi.ru>

<http://www.chereda2@yandex.ru>

Abstract. This article deals with the problem to display on the screen data base tables by means DataGridView and ComboBox controls. Development environment is Microsoft Visual Studio, programming language is C#. Data base tables include one to many relationships. There is consideration eight combinations pairs of DataGridView and ComboBox controls. Is shown all these cases one can realize without coding using drag and drop technology and choosing strings in different lists and windows.

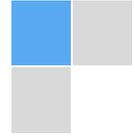
Keywords: C#, Data Base, DataGridView, ComboBox, DataGridViewComboBoxColumn, Primary Key, Foreign Key, BindingSource, Relationship, DataSource, DataSet, Smart-tag.

ISSN 2306-1561

Автоматизация и управление в технических системах (АУТС)

2014. – №2. – С. 41-59.

DOI: 10.12731/2306-1561-2014-2-5



УДК 681.3.07

Отображение таблиц баз данных на экране

Чернов Эдгар Александрович

Российская Федерация, кандидат технических наук, доцент кафедры «Автоматизированные системы управления».

ФГБОУ ВПО «Московский автомобильно-дорожный государственный технический университет (МАДИ)», 125319, Российская Федерация, г. Москва, Ленинградский проспект, д.64, Тел.: +7 (499) 151-64-12, <http://www.madi.ru>

<http://www.chereda2@yandex.ru>

Аннотация. Настоящая статья посвящена обзору вариантов отображения (без кодирования) на экране реляционных таблиц, связанных отношением один ко многим, с помощью элементов управления DataGridView и ComboBox в среде Visual Studio 2012.

Ключевые слова: язык С#, визуальное программирование, реляционная связь, первичный ключ, внешний ключ, отношение один ко многим, родительская таблица, дочерняя таблица, источник привязки, отображение на экране.

1. Введение

Большинство современных приложений обращается к базам данных для получения информации. Проблема обеспечения удобного и быстрого доступа к данным и адекватное представление ее на экране становится все более актуальной по мере увеличения объемов информации, хранящейся в базах данных [1 – 15].

Элементы управления DataGridView и ComboBox и некоторые другие предназначены для отображения таблиц на экране. Информация в базах данных представляется в виде таблиц. Для сокращения представляемых таблиц применяются фильтры. Ускоряет также просмотр представление информации в виде иерархических структур. В частности, всплывающие списки (элемент управления ComboBox) позволяют обеспечить иерархический интерфейс при просмотре таблиц, при котором всплывающий список выбирает таблицы или части таблиц.

Создание иерархических моделей представления информации всегда можно выполнить программно, но в современных средах Microsoft Visual Studio это можно реализовать визуально, перетаскивая таблицы на экран и выбирая соответствующие

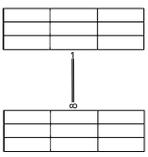
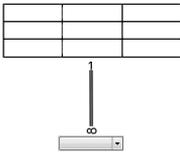
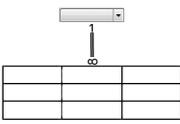
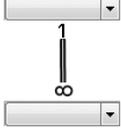
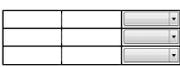
компоненты в окнах диалога и меню. Такой подход существенно сокращает время на разработку приложений, исключая ошибки кодирования.

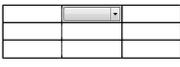
Судя по имеющейся в Интернете информации, разработчики зачастую не знают визуальных методов отображения таблиц. Данная статья посвящена анализу сочетаний таблиц и всплывающих списков при построении иерархических структур представления информации, их возможностям и способам визуальной реализации (без кодирования). Рассмотрены пары таблиц: родительская → дочерняя. К ним в любом направлении могут быть добавлены таблицы с реляционными отношениями.

Рассмотренные ниже примеры реализованы в среде языка C#, но могут быть использованы и другие языки. Все операции реализованы с помощью окон диалога, вызываемых при щелчке по смарт-тегу - значку , появляющемся на правом верхнем углу выбранного элемента управления.

2. Ассоциации списков и таблиц

Ниже представлены некоторые комбинации всплывающих списков (элемент управления ComboBox) и таблиц (элемент управления DataGridView). Рассматривается реляционное отношение «один ко многим». Сверху размещен родительский элемент.

1.		<p>Две таблицы, связанные отношением первичного и внешнего ключей в режиме «один ко многим», позволяющее реализовать листание в режиме: «Главная таблица» → «Детализация».</p>
2.		<p>Таблица и всплывающий список, связанные по внешнему ключу, позволяющие реализовать листание в режиме: «Главная таблица» → «Детализация», если в дочернем списке отображается только один столбец. С помощью независимого ComboBox можно переключать отображаемые столбцы по индексу столбца в таблице.</p>
3.		<p>Всплывающий список представляет родительскую таблицу, а элемент DataGridView отображает дочернюю таблицу. Например, в списке указаны товары, а в таблице их характеристики.</p>
4.		<p>Реляционная связь строк всплывающего списка с дочерним всплывающим списком. Например, в главном списке выбирается страна, в дочернем списке поставщики (в таблице, привязанной к дочернему списку, например, товар).</p>
5.		<p>Таблица, столбец которой состоит из всплывающих списков (DataGridViewComboBoxColumn): Все всплывающие списки в столбце имеют одинаковое содержимое.</p>

		Каждый из всплывающих списков имеет разное содержимое, например, список абонентов и их телефоны (домашний, рабочий, мобильный).
6.		Столбцы типа DataGridViewComboBoxColumn, связанные в каждой строке отношением один ко многим.
7.		Всплывающий список в заголовке столбца, обеспечивающий сортировку и фильтрацию по выбранному в списке значению.
8.		Всплывающий список, каждая строка которого привязана к отдельной таблице. Выбор строки в ComboBox переключает таблицу. Например, в ComboBox перечислены города, а таблицы содержат данные по поставщикам.

Указанные комбинации списков можно рассматривать в качестве основных, и они могут представлять собой элементы более сложной структуры.

Для реализации отображений на экране перечисленной комбинации списков на элементах управления (DataGridView или ComboBox) достаточно реализовать следующие отображения:

- отображение родительской (главной) таблицы из базы данных;
- отображение дочерней таблицы (подробностей) из базы данных. Это отображение реализует реляционную связь между таблицами.

Любое из указанных отображений может быть узлом в иерархии привязок.

3. Подключение к приложению источников данных

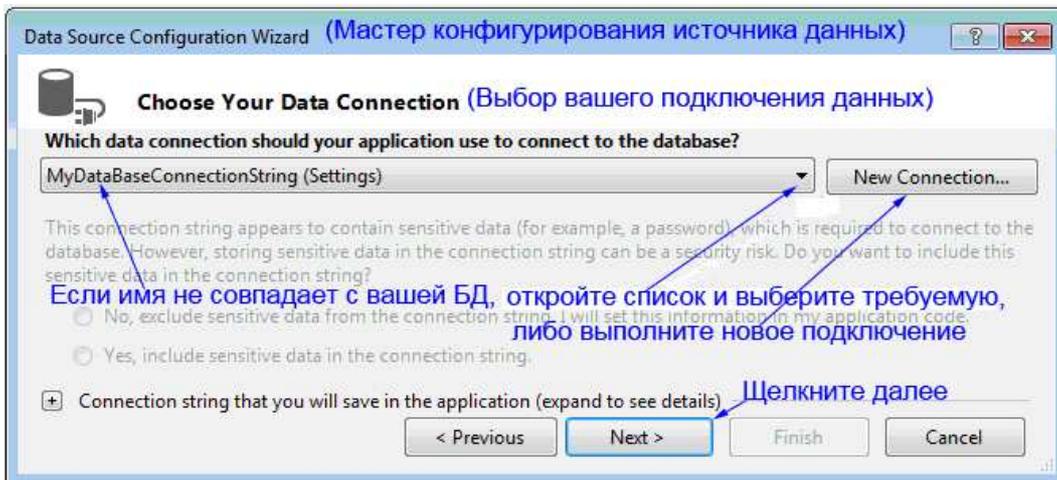
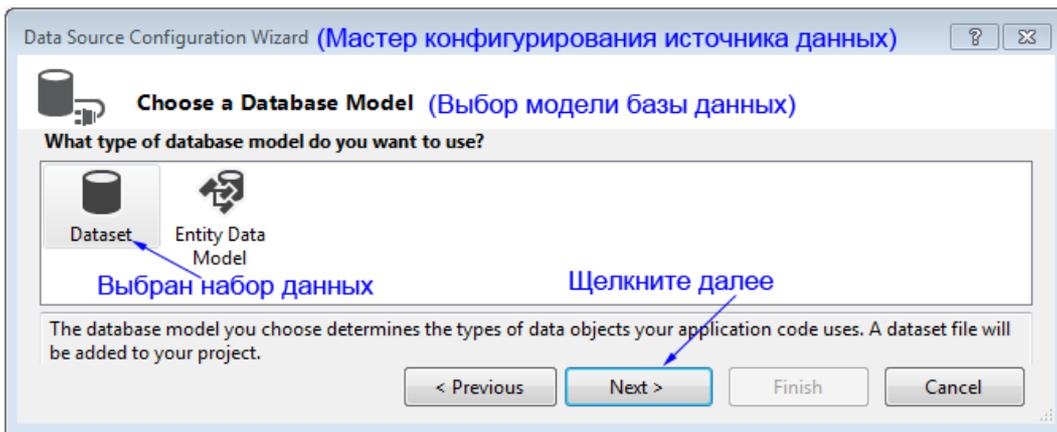
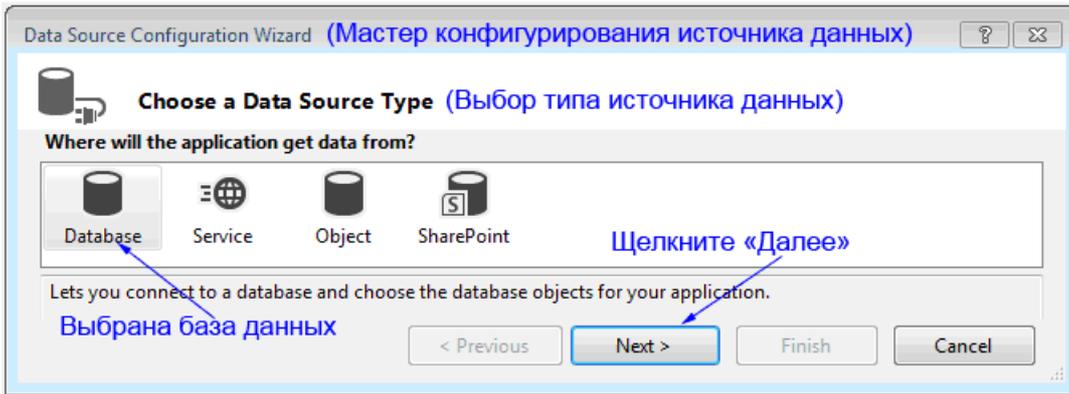
Относительно рассматриваемых отображений сделаны предположения:

- источниками данных для приложения являются таблицы, хранящиеся в базе данных;
- между таблицами в базе данных реализованы реляционные связи.

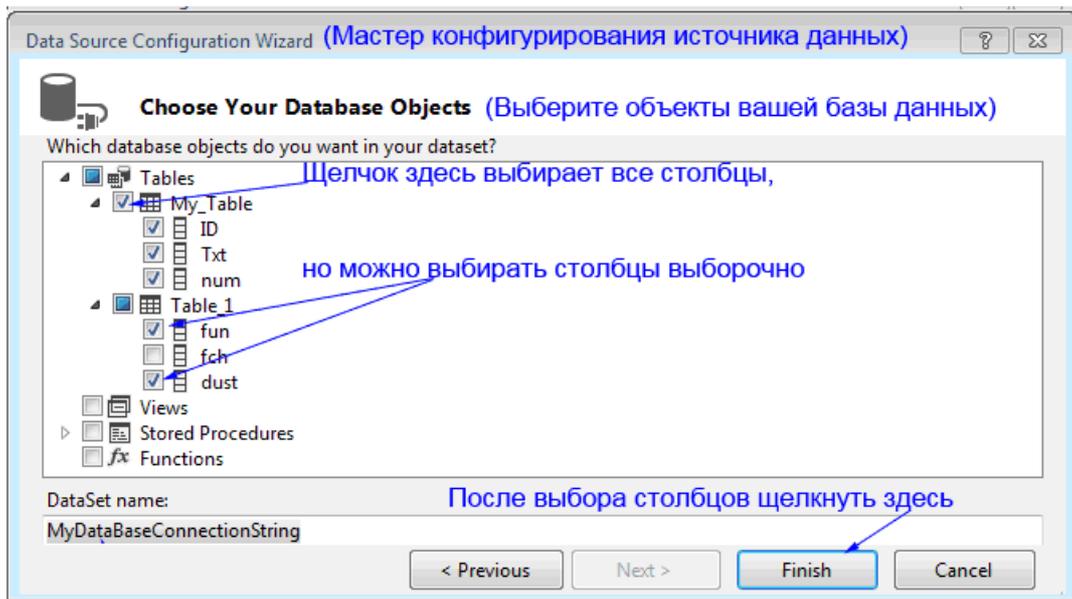
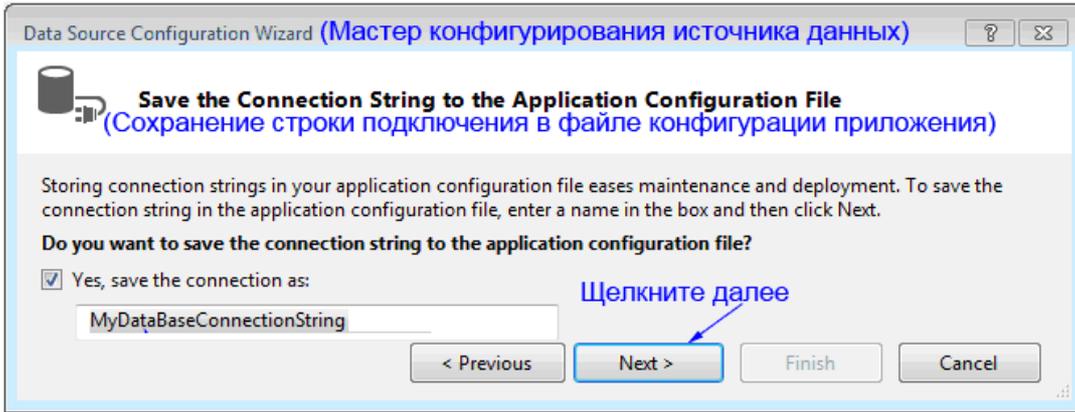
Для подключения базы данных к приложению в проекте Windows Forms Application проделать следующее:

1. В среде Microsoft Visual Studio запустите проект Windows Forms Application для языка C#
2. Войдите в окно конструктора форм (Form1[Design]).
3. Перейдите в окно «Источники данных» (Data Sources). Если это окно на экране отсутствует, откройте на главной панели меню «Вид» (View), выберите позицию «Другие окна» (Other Windows) и щелкните по позиции «Источники данных (Data Sources)».

Щелкните по надписи «Добавить Новый Источник Данных» (Add New Data Source) для вызова «Мастера конфигурирования наборов данных» (Data Source Configuration Wizard). Выполните проход по окнам (скриншоты окон уменьшены по высоте):



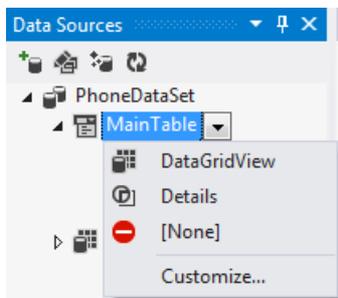
На этом экране проверьте, чтобы в строке соединения с базой данных имя совпадало с именем вашей базы данных (в данном примере имя MyDataBase). Если это не так, щелкните по кнопке Новое подключение (New Connection), и подключитесь к требуемой базе данных.



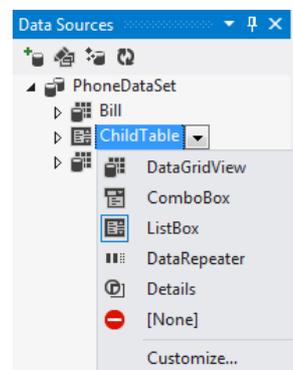
4. Выбранные таблицы будут добавлены в окно «Источников данных» (Data Sources).

4. Отображение таблиц на экране

Ниже показана таблица базы данных со вскрытым списком элементов для отображения таблицы на экране:



Сокращенный список



Полный список

В примере справа перечислены все возможные варианты (обычно в списке в наличии элементы, показанные слева. Элемент DataGridView может быть заменен). Выбор элементов в списке для отображения задается в позиции Customize. Один из элементов (в данном примере DataGridView) задан в качестве элемента, выбираемого по умолчанию.

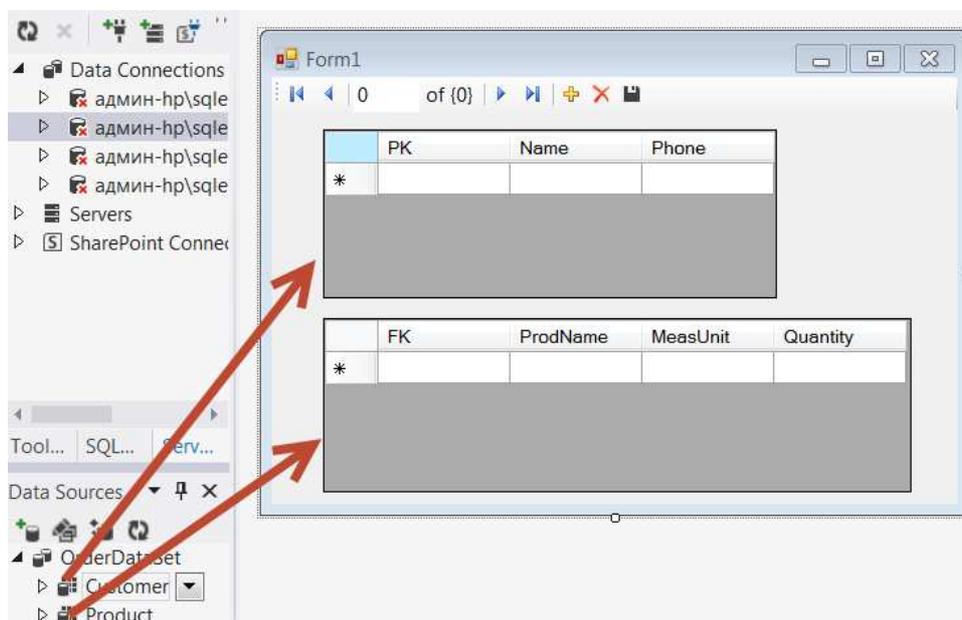
Независимо от типа таблицы для отображения на экране, всегда можно выбрать, на каком элементе управления она может быть отображена.

Таблицы можно перетаскивать на форму (в режиме конструктора Form1[Design]) или привязывать к источнику данных через смарт-тег.

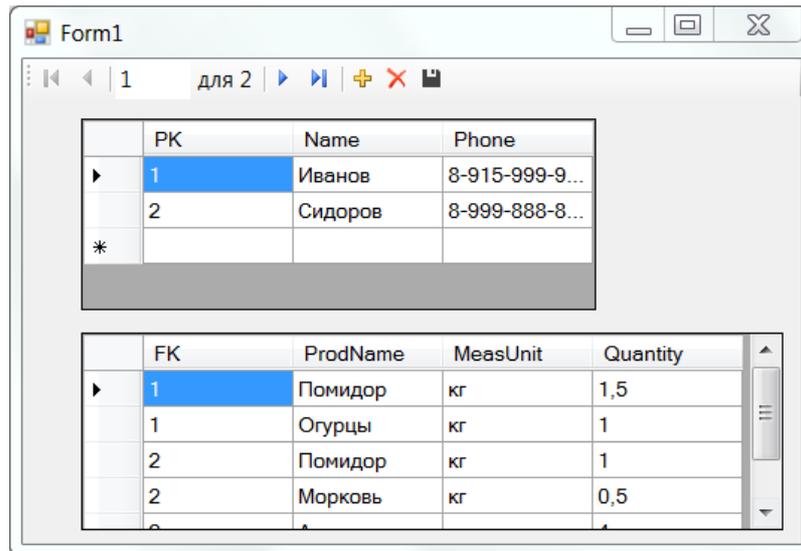
4.1. Отображение на экране родительских и автономных таблиц

Родительские и дочерние таблицы (реляционные) хранятся в базах данных. Автономные таблицы могут размещаться как в базах данных, так и быть локальными. Ниже рассмотрены таблицы из баз данных. Таблицы могут отображаться как в таблицах (элемент DataGridView), так и во всплывающих списках (элемент ComboBox). Если подключена база данных, содержащая таблицы, например, Customer и Product, тогда для отображения их на экране сделайте следующее:

1. Войдите в окно «Источники данных» (Data Sources), захватите мышью таблицу Customer и перетащите ее на форму. Затем аналогично перетащите на форму таблицу Product



2. После перетаскивания таблиц система для вставленных элементов управления DataGridView присвоит имена
customerDataGridView – для верхней таблицы
productDataGridView – для нижней таблицы
3. Кроме того, под окном формы будут вставлены адаптеры таблиц, менеджер таблиц и привязки (customerBindingSource и productBindingSource).
4. После запуска приложения таблица будет иметь, например, такой вид:



Вам будет также добавлен навигатор для перемещения по строкам таблицы и кнопки для сохранения таблицы, а также для добавления и исключения строк. Для исключения строки (в запущенном приложении) ее надо выбрать и щелкнуть по кнопке . В дочерней таблице строки можно исключать независимо от главной таблицы, вплоть до полного исключения строк. Но при исключении строки в главной таблице исключаются все строки в дочерней таблице. Строки исключаются на экране, но из базы данных они будут исключены только после сохранения изменений.

Добавляемая (пустая) строка (кнопка ) вставляется внизу главной таблицы, и ее требуется заполнить. Первичный ключ сначала получает отрицательное значение, а после завершения оформления строки (кнопка  - сохранение изменений) значение (положительное) первичному ключу назначает система. Вопросы разрешения и запрещения корректировки таблиц здесь не рассматриваются.

При перетаскивании на форму первой таблицы все отображенные в источнике данных таблицы переносятся в отсоединенный набор данных DataSet. Кнопки навигатора позволяют управлять всеми таблицами, но в каждый момент времени относятся к выбранной таблице.

4.2. Отображение на экране дочерних таблиц

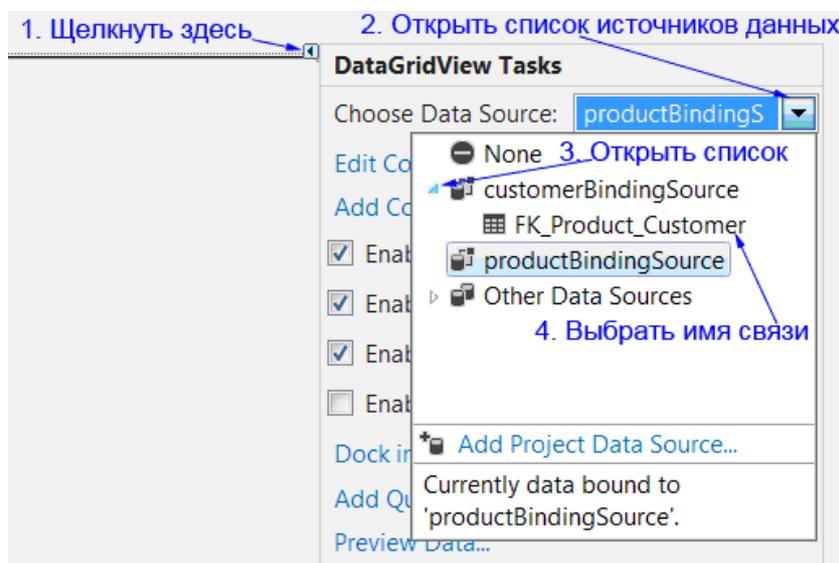
Дочерние таблицы могут отображаться как в таблицах (элемент DataGridView), так и во всплывающих списках (элемент ComboBox).

Отображение дочерней таблицы на DataGridView

Отображаемая дочерняя таблица должна быть перетащена на форму и для отображения должен быть выбран элемент DataGridView.

1. В окне конструктора (Form1[Design]) щелкните по таблице Product (дочерней, нижней).

2. Щелкните по смарт-тегу (откройте список DataGridView Tasks).
3. Откройте список источников данных.
4. Откройте список привязок родительской таблицы customerBindingSource.
5. Выберите имя связи и щелкните по нему.

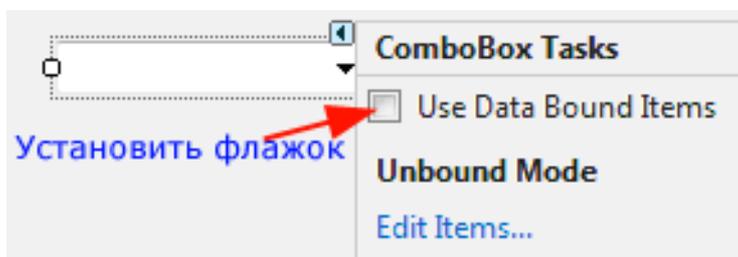


6. Проверьте работу приложения.

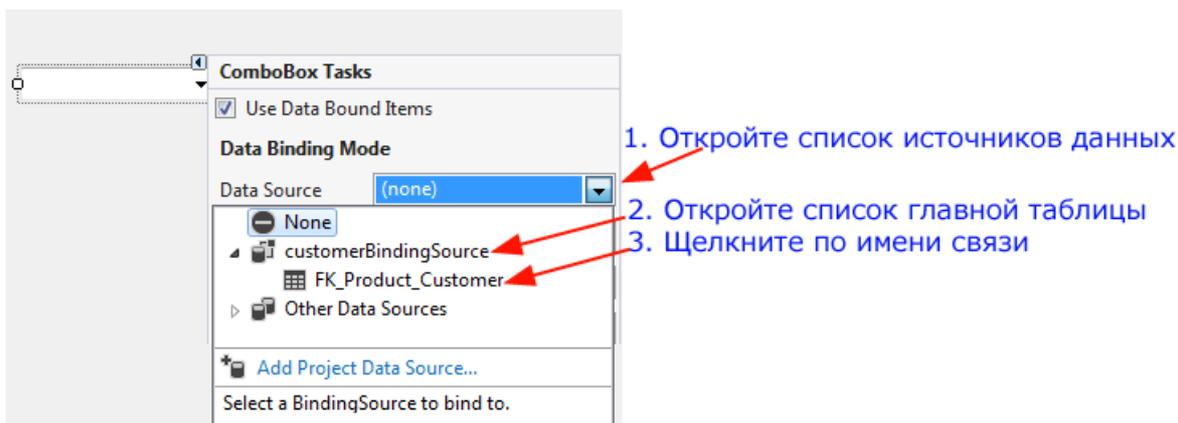
Примечание. Подключить таблицу из базы данных можно и через смарт-тег, но если это первая таблица, то не будет добавлен на форму навигатор. В этом случае сохранение изменений в базе данных станет проблематичным.

Отображение дочерней таблицы на ComboBox

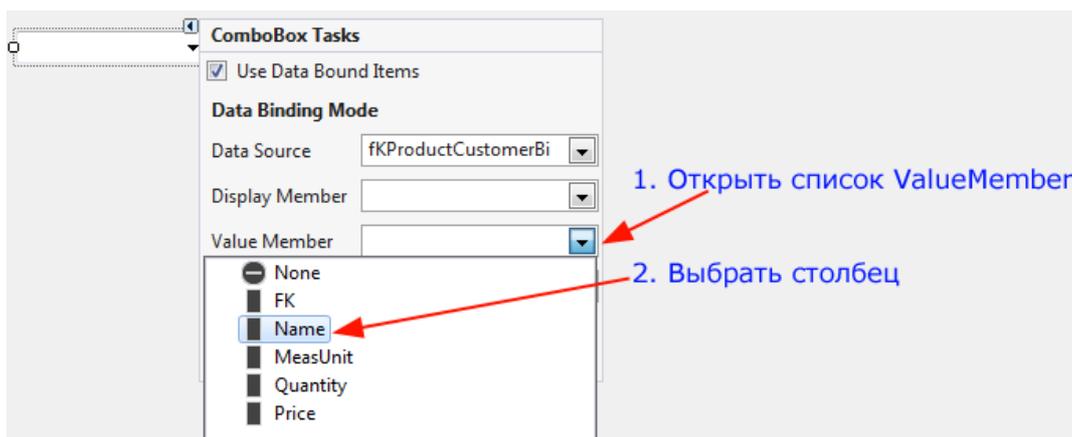
1. Находясь в окне конструктора (Form1[Design]) выберите таблицу, откройте список элементов для отображения (см. параграф 4) и выберите ComboBox.
2. Щелкните по смарт-тегу элемента ComboBox.
3. Установите флажок в CheckBox для строки «Использовать элементы привязки данных» (Use Data Bound Items).



4. Откройте список источников данных



5. Откройте список связей главной (родительской таблицы) таблицы.
6. Щелкните по имени связи. (Имя связи изменит форму записи, станет: fkProductCustomerBindingSource) и будет помещено в список источников данных.
7. Откройте список ValueMember. (Только этот элемент может хранить одинаковые величины, которыми являются внешние ключи – FK)



8. Выберите столбец для отображения (в ComboBox можно отобразить только один столбец). Компилятор автоматически определяет внешний ключ, который задан связью (в ComboBox этот ключ сохраняется в ValueMember).

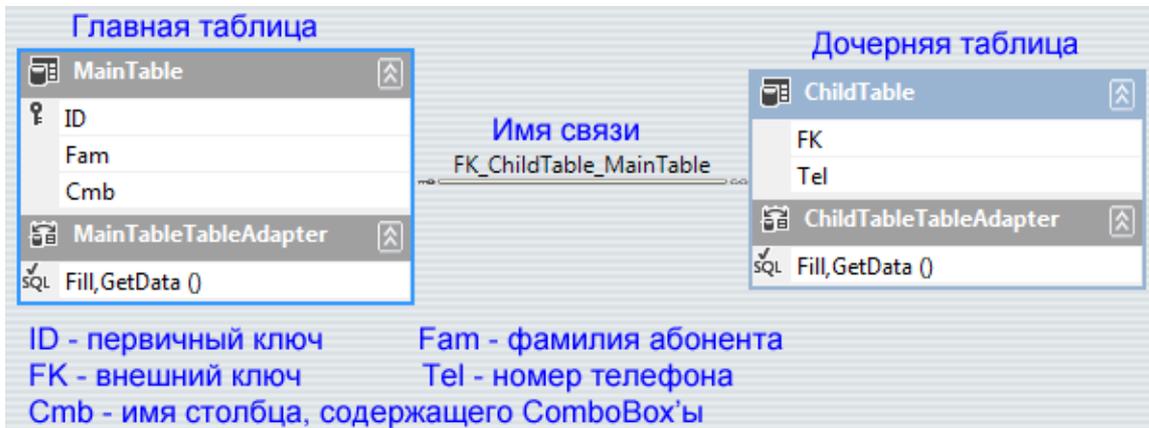
Показанные выше способы отображения на экране родительских и дочерних таблиц позволяет реализовать первые четыре комбинации DataGridView и ComboBox. Для этого при перетаскивании на экран требуется выбрать требуемый элемент управления и учесть особенность отображения таблицы на элементе ComboBox.

5. Столбец таблицы, состоящий из ComboBox

Столбец таблицы элемента DataGridView может содержать в качестве ячеек всплывающие списки. Содержимое ComboBox может либо быть одинаковым во всех ячейках (например, список существующих товаров), либо различающимся в каждой ячейке (например, список абонентов, у каждого из которых несколько телефонов).

Источником данных для ComboBox может быть либо автономная таблица в базе данных (одинаковые списки у всех ComboBox), либо дочерняя таблица (разные списки).

Ниже представлена схема базы данных (для отображения телефонов для абонентов).



При использовании таблиц, содержащих столбец всплывающих списков, возникают перечисленные ниже побочные явления.

- Список в столбце таблицы вскрывается после второго щелчка (первый щелчок – выбор ячейки, второй – вскрытие списка). Это неудобно. Исправить этот недостаток можно, например, так:
 1. Добавьте обработчик события CellEnter.
 2. В обработчик вставьте приведенный ниже код.

```
private void mainTableDataGridview_CellEnter(object sender,
DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == ind)
    {
        mainDGV.EditMode = DataGridViewEditMode.EditOnEnter;
    }
}
```

В примере кода: mainTable - имя главной (родительской) таблицы в наборе DataSet приложения, mainDGV - имя элемента DataGridView, ind - номер столбца DataGridView, состоящего из ComboBox.

- При запуске приложения иногда возникает исключительная ситуация, связанная с преобразованиями типов данных. Избавиться от этого можно, добавив для элемента DataGridView обработчик события DataError. В обработчик ничего вставлять не надо. Но исключительная ситуация больше появляться не будет.

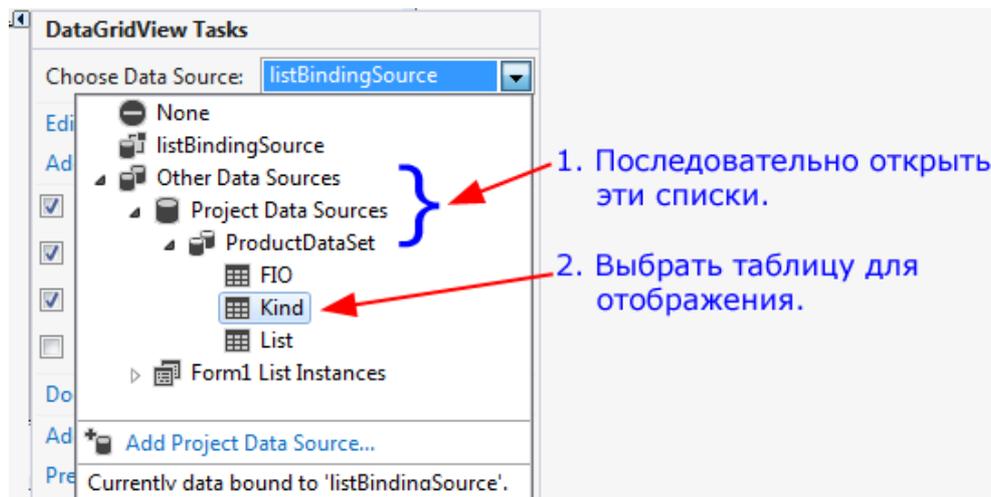
В зависимости от содержимого списков (одинаковые или разные) действия далее различны.

5.1. Столбец из ComboBox с одинаковым содержимым

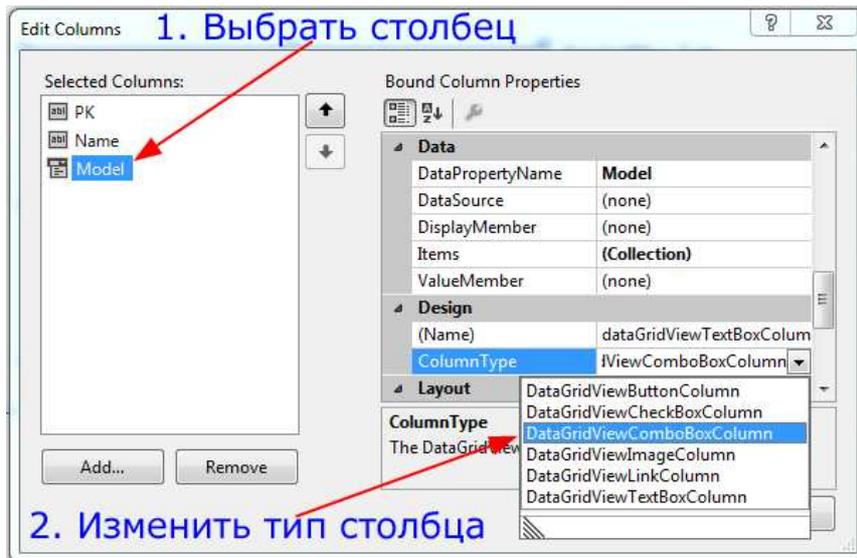
В этом случае источником данных является автономная таблица базы данных. (если это родительская таблица, она не должна быть основной). Эта таблица отсутствует в списке источников данных, поскольку на DataGridView уже отображена родительская таблица (содержащая фамилии абонентов). Ее надо добавить в список источников данных для текущей таблицы.

Для создания столбца в элементе DataGridView, состоящего из всплывающих списков с одинаковым содержимым, сделайте следующее.

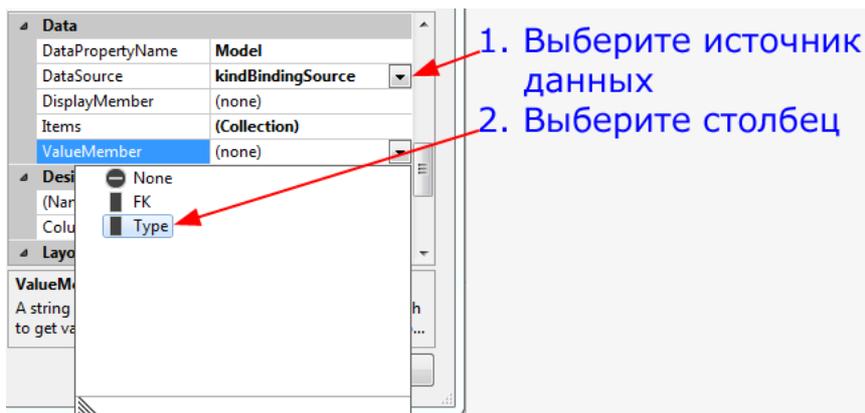
1. Создайте в среде Visual Studio проект и подключите базу данных.
2. Перетащите на форму родительскую таблицу (MainTable), выбрав элемент для отображения DataGridView.
3. Щелкнув по смарт-тегу этого элемента, откройте окно задач для элемента DataGridView и войдите в режим «Редактирования столбцов» (Edit Columns).
4. Откройте список «Других источников данных» (Other Data Sources) и вложенные списки до списка ProductDataSet. Выберите таблицу для отображения. На представленной схеме: FIO – главная родительская таблица; List – дочерняя таблица, которая будет содержать элемент ComboBox; Kind – таблица (автономная), которую будут отображать элементы ComboBox.



5. Обычно после щелчка по выбранной таблице на экране исчезает один из столбцов, поскольку был выбран другой источник данных. Для возврата его на экран выберите в списке привязку к таблице, в которую вставляется столбец из ComboBox (в данном примере listBindingSource), или просто нажмите на клавиши Alt + Backspace (это операция Undo).



6. В окне DataGridView Tasks, выбрав режим редактирования столбцов, выберите столбец, в который будут вставляться элементы ComboBox (он должен быть типа DataGridViewComboBoxColumn).



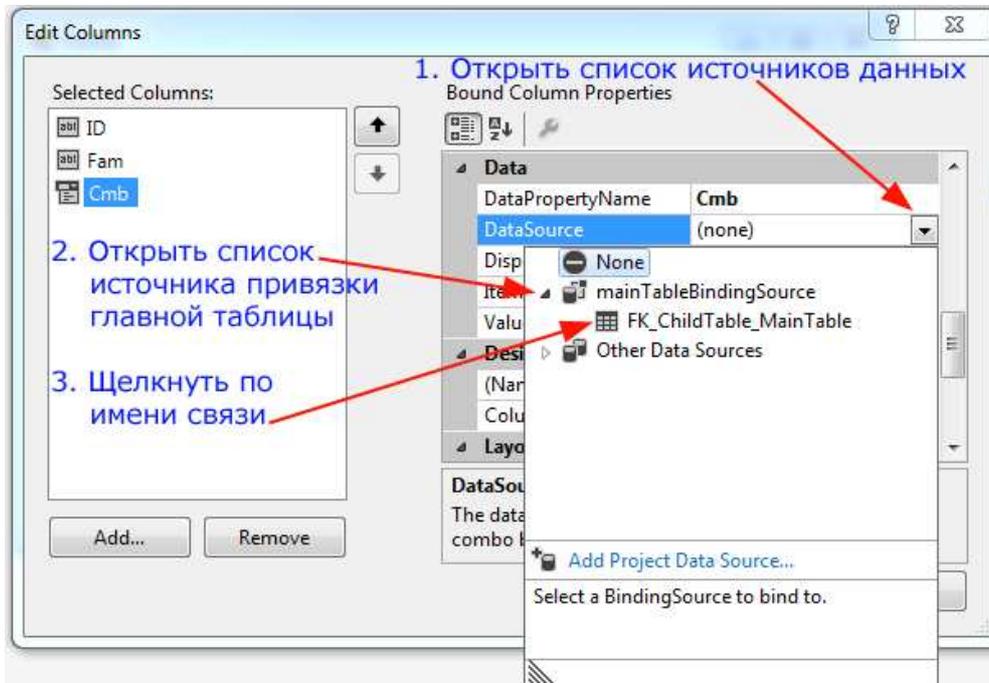
7. Выберите привязку kindBindingSource в качестве источника данных (Data Source).
8. Для отображаемого свойства DisplayMember выберите столбец Type.
9. Проверьте работу приложения и устраните недостатки, как указано в параграфе 5.

5.2. Столбец, состоящий из ComboBox с разным содержимым

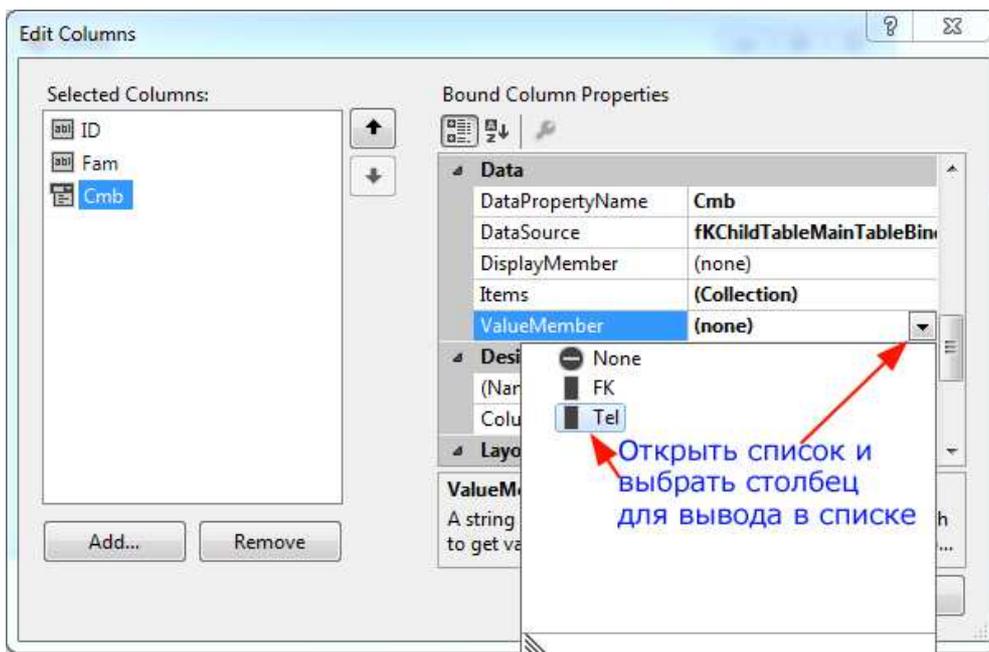
Для получения разных списков в каждом из ComboBox требуется, чтобы эти списки были в дочерней таблице. Если родительская таблица содержит фамилии абонентов, то дочерняя таблица может содержать список телефонов для каждого абонента. А этого можно достичь, если выбрать в качестве источника данных имя реляционной связи между таблицами.

1. Создайте в среде Visual Studio проект и подключите базу данных.
2. Перетащите на форму родительскую таблицу (MainTable), выбрав элемент для отображения DataGridView.

- Щелкнув по смарт-тегу этого элемента, откройте окно задач для элемента DataGridView и войдите в режим «Редактирования столбцов» (Edit Columns).
- Выберите соответствующий столбец и в списке «Типов столбцов» (ColumnType) выберите тип DataGridViewComboBoxColumn (см. пункт 6 в параграфе 5.1).



- Откройте список источников данных, откройте список источника привязки главной (родительской) таблицы и щелкните по имени связи.



- Откройте список ValueMember и выберите имя столбца для отображения во всплывающем списке. Позиция DisplayMember не выбирается. Это связано с

тем, что только свойство ValueMember может иметь значение (повторяющееся) внешнего ключа, а что будет выводиться – это вторично.

5.3. Два столбца, состоящие из ComboBox

Автономные таблицы можно отображать на любом количестве столбцов. Требование одно: столбец должен быть типа DataGridViewComboBoxColumn (Как отобразить один столбец, описано в параграфе 5.1).

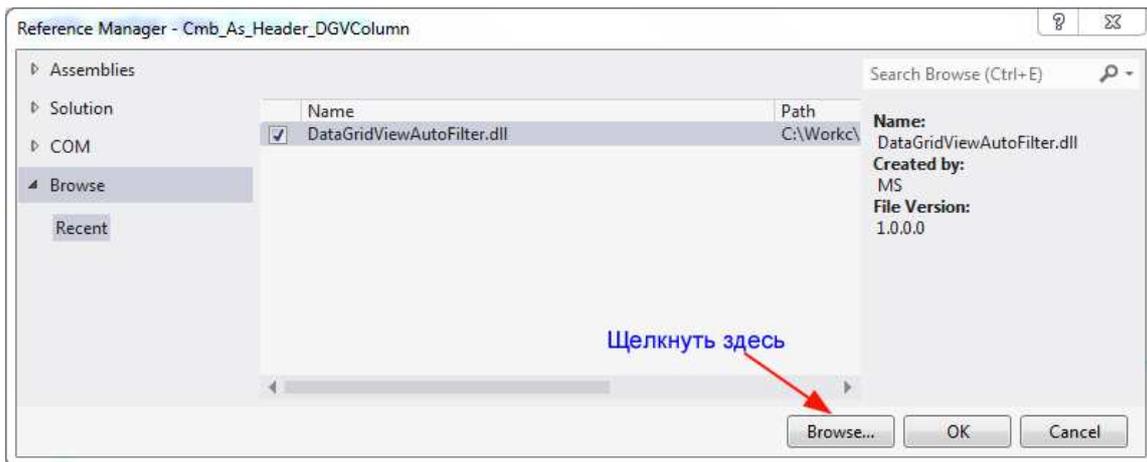
Возможно отображение двух столбцов, состоящих из элементов ComboBox, и связанных реляционным отношением. В этом случае связь реализуется между двумя элементами ComboBox, размещенными в одной строке. Выбор позиции в одном элементе задает содержимое связанного с ним элемента.

В этом случае для каждой пары столбцов должна существовать отдельная связка из родительской и дочерней таблиц. Сначала требуется отобразить родительскую таблицу (см. параграф 5.1), а затем дочернюю (см. параграф 5.2).

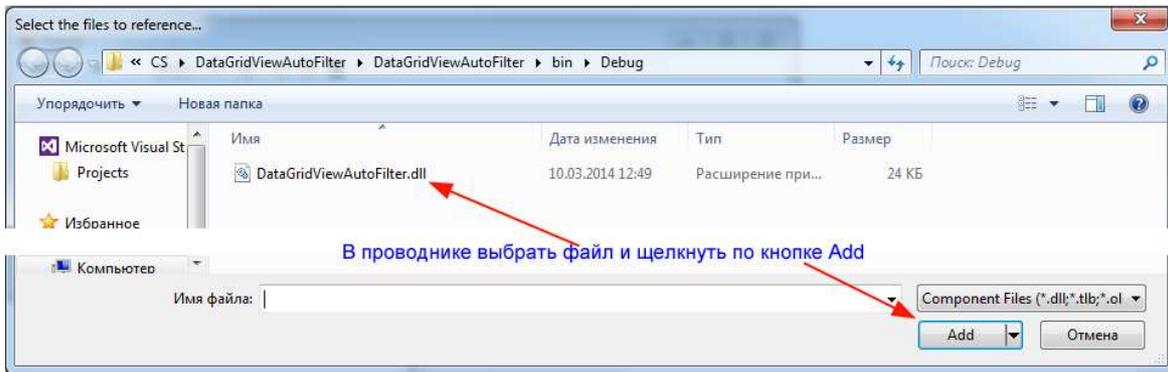
6. Всплывающий список в заголовке столбца

С помощью всплывающего списка в заголовке столбца может реализовать фильтрацию по выбранному значению. Для этого требуется подключить библиотеку DataGridViewAutoFilter.dll. Эта библиотека не является сборкой, поэтому ее надо преобразовать. Прделайте следующее.

1. В «Обозревателе решений» (Solution Explorer) щелкните правой кнопкой по позиции «Ссылки» (References) и в появившемся меню выберите позицию «Добавить ссылку» (Add Reference).
2. В окне диалога (Reference Manager) щелкните по нижней кнопке «Обзор» (Browse).



3. Появится приведенное ниже окно. В проводнике (через список «Компьютер») выбрать файл и щелкнуть по кнопке «Добавить» (Add).

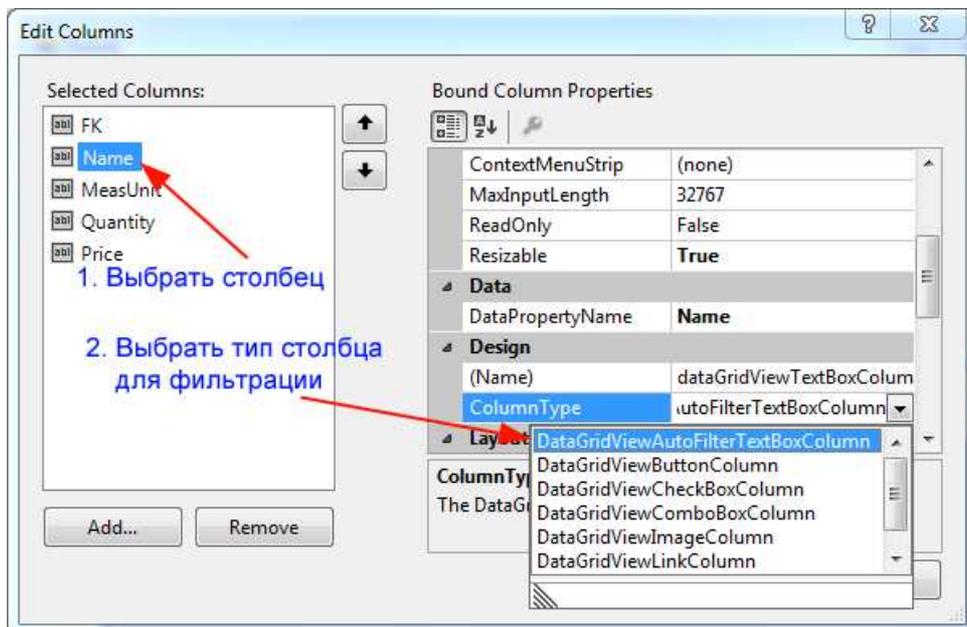


4. Будет выполнен возврат в окно, показанное в пункте 2. Щелкните по кнопке ОК. В окно будет добавлена ссылка на сборку.
5. Теперь этот файл можно добавить к проекту как сборку:

```
using DataGridViewAutoFilter;
```

Вставка элемента ComboBox в заголовок столбца (фильтрация)

1. Подключите к проекту базу данных.
2. Подключите к проекту сборку DataGridViewAutoFilter.
3. Перетащите на форму требуемую таблицу.
4. Откройте список задач элемента DataGridView. (щелкнув по смарт-тегу).
5. Войдите в режим редактирования столбцов и выберите столбец, для которого будет подключена фильтрация (и сортировка).

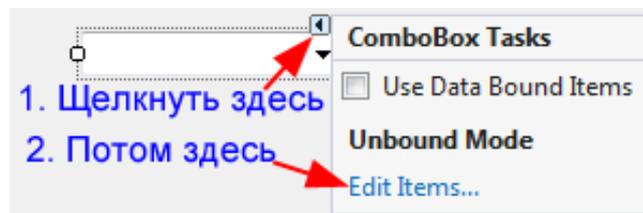


6. Проверить работу приложения. Элементы столбца будут автоматически отсортированы по алфавиту. При выборе во всплывающем списке позиции остаются только строки, содержащие это значение в выбранной позиции.

7. Элемент ComboBox выбирает таблицы

Содержимое списка (элемент ComboBox) может храниться в базе данных, но список может быть и локальным (в примере ниже рассмотрен локальный список). Для реализации такой связки проделать следующее.

1. Создать проект.
2. Подключить базу данных.
3. В конструкторе (Form1[Design]) добавить на форму элемент DataGridView. Установить для него свойство Visible в значение false, свойство для AutoSizeMode установить в значение Fill. Это требуется, чтобы при разных количествах столбцов в таблице заполнение было равномерным. Таблица будет по потребности растягиваться или сжиматься по ширине.
4. Добавить на форму элемент ComboBox.
5. Щелкнуть по смарт-тегу этого элемента и выбрать позицию Edit Items.



6. В открывшемся окне набрать строки (без кавычек), задающие таблицу (например, «Таблица 1», «Таблица 2» и т. д.).
7. Чтобы при запуске была выбрана таблица (чтобы список не был пустым), добавить в конструктор формы (после InitializeComponent();) строку:

```
comboBox1.SelectedIndex = 0;
```

8. Добавить обработчик события SelectedIndexChanged для элемента ComboBox.
9. В этом обработчике реализовать привязку таблицы к позиции всплывающего списка:

```
private void comboBox1_SelectedIndexChanged  
(object sender, EventArgs e)  
{  
    int n = comboBox1.SelectedIndex;  
    BindingSource bs = new BindingSource();  
    bs.DataSource = this.vegetablesDataSet.Tables[n];  
    dataGridView1.DataSource = bs;  
}
```

В этом фрагменте переменная n получает значение индекса выбранной строки в элементе ComboBox. Затем объявлен новый источник привязки, источником данных для которого указана таблица в наборе данных vegetablesDataSet (здесь vegetables – имя базы данных) с индексом,

выбранном в списке. Затем эта привязка назначена в качестве источника данных для таблицы (элемент `dataGridView1`).

10. Если вместо индекса таблицы требуется название из списка, то вместо `SelectedIndex` использовать `SelectedItem`. В этом случае название из списка должно совпадать с именем столбца в базе данных, а в квадратных скобках это имя должно заключаться в кавычки.

8. Заключение

Из вышеизложенного следует:

1. Таблицы баз данных можно отображать на экране без кодирования, применяя перетаскивание из окна источников данных приложения и выбирая компоненты в окне, всплывающем при щелчке по смарт-тегу элемента управления (источник данных элемента управления).
2. Для отображения родительской таблицы (элемент управления `DataGridView`) ее перетаскивают на форму в конструкторе `Form1[Design]` из окна источников данных.
3. Для отображения дочерней таблицы в источнике данных элемента управления (щелчок по смарт-тегу элемента `DataGridView` дочерней таблицы) открывают список связей родительской таблицы и выбирают требуемую.
4. При отображении таблиц на элементе управления `ComboBox` сначала устанавливают флаг признака привязки элемента к базе данных (после щелчка по смарт-тегу).
5. На элемент управления `ComboBox` родительская таблица отображается посредством выбора ее в списке доступных таблиц (см. пункт 4). Выбор столбца для отображения задается в позиции `DisplayMember`.
6. На элемент управления `ComboBox` дочерняя таблица отображается посредством выбора имени связи родительской таблицы (см. пункт 4). Выбор столбца для отображения задается в списке, появляющемся после выбора позиции `ValueMember`. Свойство `DisplayMember` в этом случае не задается.
7. Если столбец таблицы состоит из элементов `ComboBox` с одинаковым содержимым, то таблица для него выбирается из списка «Другие источники данных» (`Other Data Sources`) в окне задач для `DataGridView`.
8. Если столбец таблицы состоит из элементов `ComboBox` с разным содержимым, то таблица для него задается именем связи с родительской таблицей.
9. Для добавления элемента `ComboBox` в заголовок столбца к приложению добавляют библиотеку `DataGridViewAutoFilter.dll`. После этого выбирается тип столбца с такой структурой.
10. Для выбора таблиц с помощью элемента `ComboBox` связывают выбранную в этом элементе позицию с индексом таблицы в наборе `DataSet`.

Список информационных источников

- [1] Эндрю Троелсен. Язык программирования С# 5.0 и платформа .NET 4.5. 6-ое издание. Издательство «Вильямс»,-1913, 1312 С.
- [2] MSDN.MICROSOFT.COM Справки по ключевым словам.
- [3] Шилдт Г. С# Полное руководство. Издательство «Вильямс»,- 2011, 1056.
- [4] www.uninettunouniversity. Chapter one. 2011.
- [5] www.saintangelos.com/.../Programming_Basics 2011.
- [6] csharp.net-information.com/ С# Help
- [7] Круглов А.М. Актуализация сведений о данных информационной системы средствами активного словаря-справочника данных / А.В. Будихин, Д.А. Буров, А.М. Круглов, А.В. Остроух // Научный вестник МГТУ ГА. Серия Аэромеханика и прочность. – 2007. - №119 (9). – С. 166-171.
- [8] Николаев А.В. Принципы организации динамических интерфейсов доступа к данным с использованием словарей-справочников данных / А.В. Будихин, Д.А. Буров, А.В. Николаев, А.В. Остроух // Научный вестник МГТУ ГА. Серия Аэромеханика и прочность. – 2007. - №119 (9). – С. 172-178.
- [9] Николаев А.В. Использование словаря-справочника данных для реализации пользовательских средств обработки информации / А.В. Остроух, С.А. Будихин, А.П. Баринов, А.В. Николаев // Приборы и системы. Управление, контроль, диагностика. – М.: «Научтехлитиздат», 2008. – №3. – С. 13-15.
- [10] Пшеничный Д.А. Анализ параметров и сравнение СУБД для реализации информационного обеспечения промышленного предприятия / А.В. Будихин, А.В. Остроух, Д.А. Пшеничный // Промышленные АСУ и контроллеры. - М.: «Научтехлитиздат», 2010. - №12. - С. 7-11.
- [11] Помазанов А.В. Методика оптимизации баз данных / А.В. Остроух, А.В. Помазанов, А.И. Белоусова, А.О. Васильева // В мире научных открытий. Серия «Проблемы науки и образования». - 2012. - №12. - С.49-54.
- [12] A.V. Ostroukh, M.N. Krasnyanskiy, S.V. Karpushkin, A.D. Obukhov. Development of Automated Control System for University Research Projects // Middle East Journal of Scientific Research. 2014. Vol. 20 (12). pp. 1780-1784. DOI: 10.5829/idosi.mejsr.2014.20.12.21091.
- [13] A. Ostroukh, A. Pomazanov. Realtime Development and Testing of Distributed Data Processing System for Industrial Company // Middle East Journal of Scientific Research. 2014. Vol. 20 (12). pp. 2184-2193. DOI: 10.5829/idosi.mejsr.2014.20.12.21106.
- [14] Krasnyanskiy M.N., Karpushkin S.V., Obukhov A.D., Ostroukh A.V. Automated control system for university research projects // International Journal of Advanced Studies (iJAS). 2014. Vol. 4, Issue 1, pp. 22-26. DOI: 10.12731/2227-930X-2014-1-4.
- [15] Mikhail Nikolaevich Krasnyanskiy, Andrey Vladimirovich Ostroukh, Sergey Viktorovich Karpushkin, Artyom Dmitrievich Obukhov, Nataliya Vyacheslavovna Molotkova and Irina Vladimirovna Galygina. Electronic Document Management Systems Structure for University Research and Education // Journal of Engineering and Applied Sciences. 2014. Vol 9. Issue 5. pp. 182-189. DOI: 10.3923/jeasci.2014.182.189.