Лабораторная работа №3

Выбран язык: С#

Примечание:

Если в процессе выполнения работы реализация каких-либо действий вызовет затруднение, можете заменить эти действия другими, более для Вас понятными. **Об этом пишите в отчете**.

Задание:

# Разработка Приложения для работы с базами данных

# Лабораторная работа №3. *Разработка Приложения для работы с базой данных «Факультет»*

Цель работы: Создание небольшой реляционной базы данных (БД) и выгрузка ее в форму так, чтобы все связи изначальной базы данных были сохранены. Также необходимо создать инструменты управления выгруженными данными и создать поиск студента по заданной комбинации или по его точной фамилии.

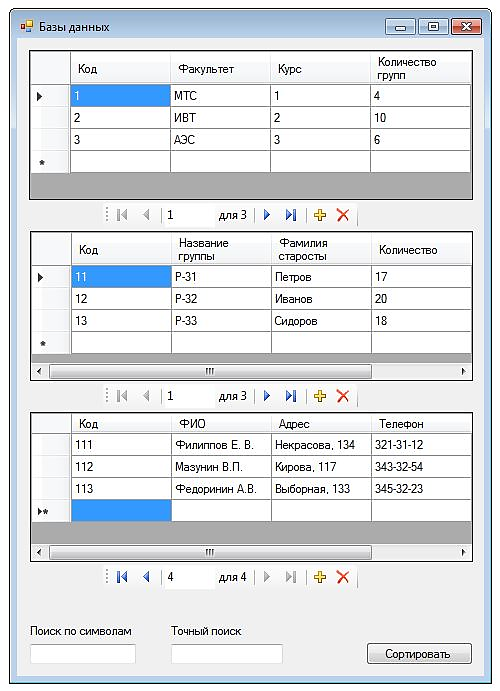


Рисунок 5. Пример интерфейса Приложения

Выполнение работы.

**Шаг 1. Создание базы данных.**

В инструкции к выполнению данной лабораторной работы рассмотрен способ создания БД через MicrosoftAccess. Все примеры и скриншоты взяты именно с него. Версия Access – 2013, VisualStudio – 2013. Вы можете применять БД любого другого формата, в т.ч. использовать инструмент для создания БД, встроенный в среду VisualStudio.

Задание: База данных должна содержать в себе три таблицы: факультеты, группы, студенты и иметь следующий тип связей:

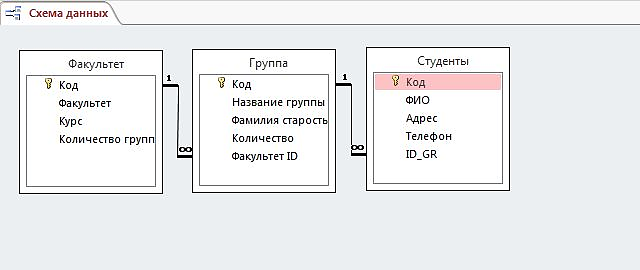


Рисунок 6. Таблицы БД с указанием связей

**Выполнение:**

Для начала создадим новый пустой проект в Access. По заданию требуется создать 3 таблицы. Одна таблица уже задана по умолчанию. Добавляем еще 2: заходим в пункт «СОЗДАНИЕ» и выбираем 2 раза «Таблица».

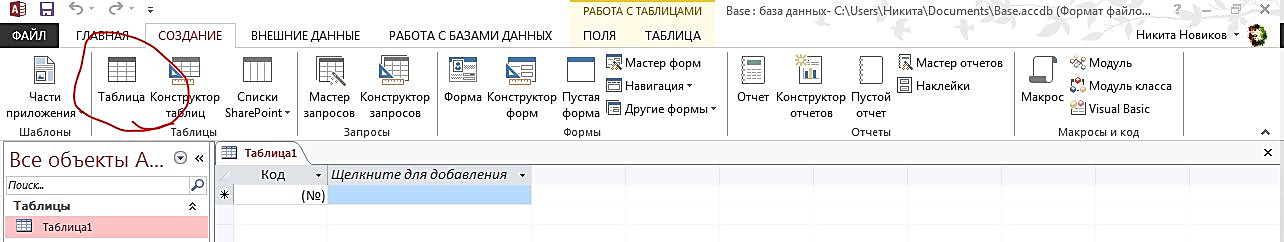
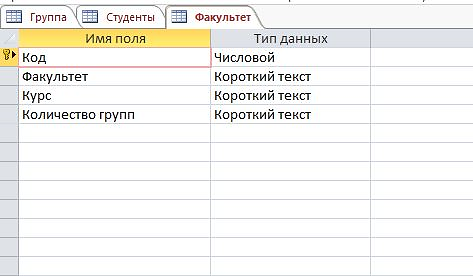
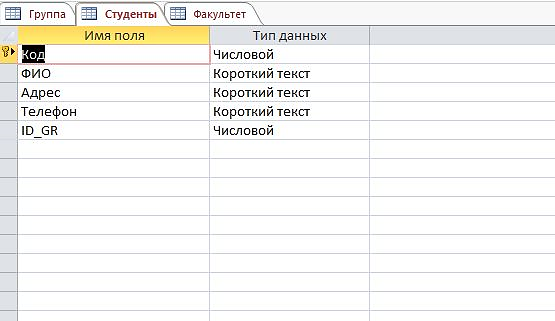


Рисунок 7. Создание таблицы

Далее обработаем каждую из них. Нужно переименовать их и добавить требуемые поля. Для этого кликаем правой кнопкой мыши по таблице и выбираем «Конструктор». Предлагается ввести имя таблицы. Первую называем «Факультет», вторую «Группа», третью – «Студенты». Далее открывается конструктор полей. Здесь нужно добавить соответствующие поля для каждой из трех таблиц (см. рисунок 5). Тип данных можно оставить по умолчанию (Короткий текст), кроме нескольких полей: «Код» (для всех трех таблиц), «Факультет ID» (для таблицы Группа) и «ID\_GR» (для таблица Студенты) – там ставим тип «Числовой». Нужно это для того, чтобы в дальнейшем создать связи между этими полями. Так как для того, чтобы создать связи без потери данных (см. Рисунок 8), необходимо, чтобы поля, между которыми будет связь, имели **один** тип данных.





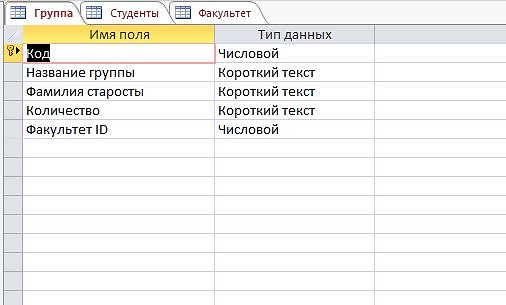


Рисунок 8. Типы данных полей таблиц

После создания полей закрываем все 3 конструктора, сохраняя все результаты и переходим к созданию связей.

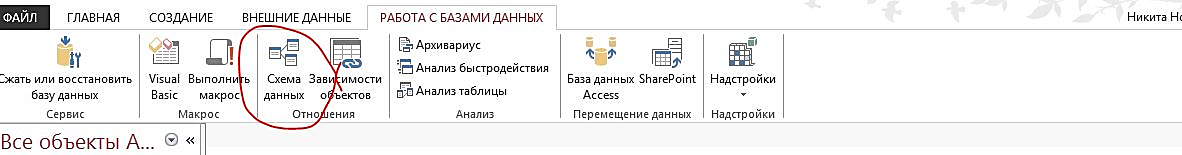


Рисунок 9. Пункт меню для организации связей между таблицами

После нажатия на «Схема данных» откроется пустое поле и появится окно, которое запросит, какие таблицы нужно поместить в это поле для создания связей. Добавляем все три таблицы и закрываем окно. Далее хватаем поле «Код» у таблицы «Факультет» и тащим его на поле «Факультет ID» таблицы «Группа». Открывается менеджер связей:

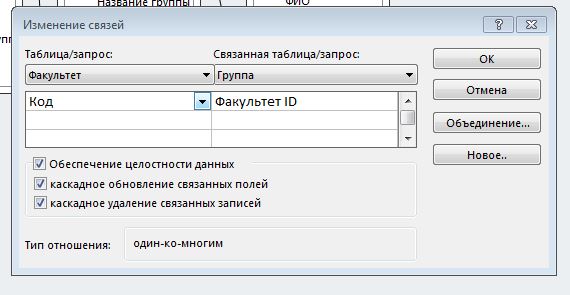


Рисунок 10. Менеджер связей.

Выставляем все галочки и нажимаем «ОК». После чего проделываем такую же операцию с таблицей «Группа» и «Студенты». Результат показан на рисунке 5. После достигнутого результата сохраняем базу и выходим из Access.

**Шаг 2. Выгрузка таблиц базы данных в форму.**

Создаем новый проект VisualStudio (язык C#) и помещаем на форму элемент DataGridView.

Примечание: в данной лабораторной работе описывается выгрузка базы данных в форму с помощью автоматической привязки базы. Однако этот способ имеет множество недостатков, например, очень сложно отвязать базу обратно, также сложно обновить привязанную базу в случае внесения Вами в нее каких-либо изменений. Для того чтобы избежать всех этих недостатков, базу можно привязывать вручную? используя заполнение таблиц с помощью драйвера OLEDB. Подробную информацию об этом можно прочитать в MSDN.

После добавления DataGridView нужно привязать нашу базу данных.

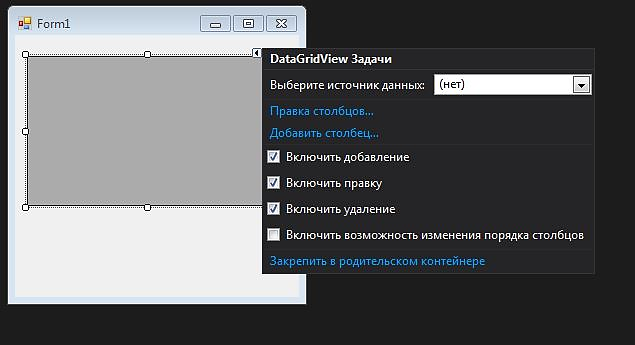


Рисунок 11. Настройка DataGridView

Т.к. источника данных пока еще нет для привязки БД, его нужно добавить. Открываем выпадающее меню пункта «Выберите источник данных» и щелкаем «Добавить источник данных проекта…»

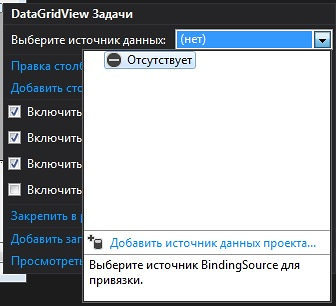


Рисунок 12. Выбор источника данных для отображения в DataGridView

Нажимаем и получаем следующее окно:

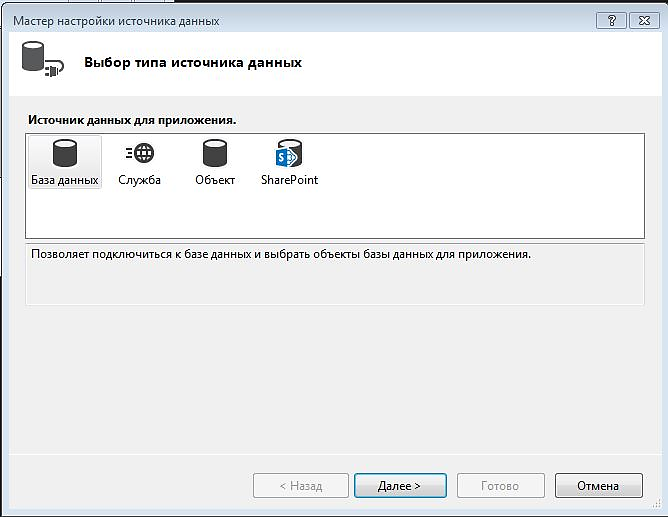


Рисунок 13. Выбор типа источника данных для отображения в DataGridView

Коли нас интересуют базы данных, их и выбираем.

Щелкаем «Далее» пару раз пока не наткнемся на вопрос «Какое подключение ваше приложение должно использовать для работы с базой данных?», где мы нажимаем на кнопку «Создать подключение…» и в появившемся окне с выбором источника данных выбираем «Файл базы данных MicrosoftAccess». Щелкаем «Продолжить» (см. Рисунок 14).

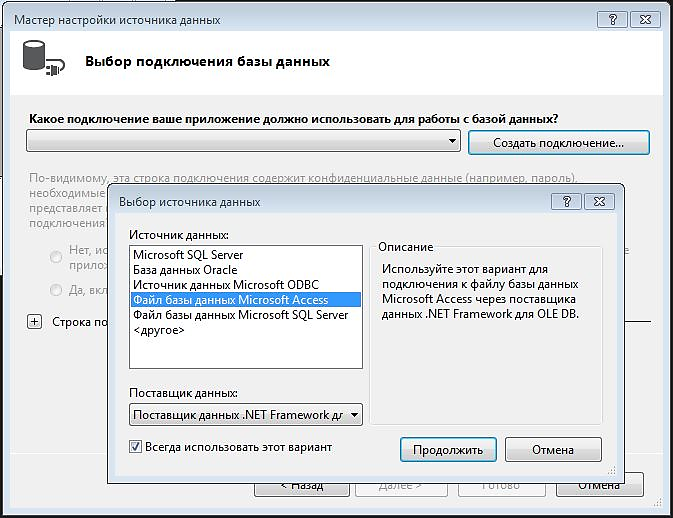


Рисунок 13. Выбор подключения для работы с базой данных

Теперь нужно указать путь на саму базу данных, которую мы создали на первом шаге. За это действие отвечает следующая кнопка:

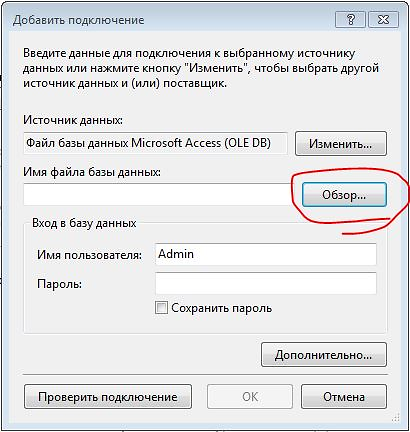


Рисунок 14. Создание подключения для работы с базой данных

После создания подключения нажимаем «Ок» и возвращаемся в форму мастера настройки источника данных. Там нажимаем «Далее», честно соглашаемся с внезапно появившимся окном о том, что «*выбранное подключение использует локальный файл данных»* и наконец то нам дают доступ к базе:

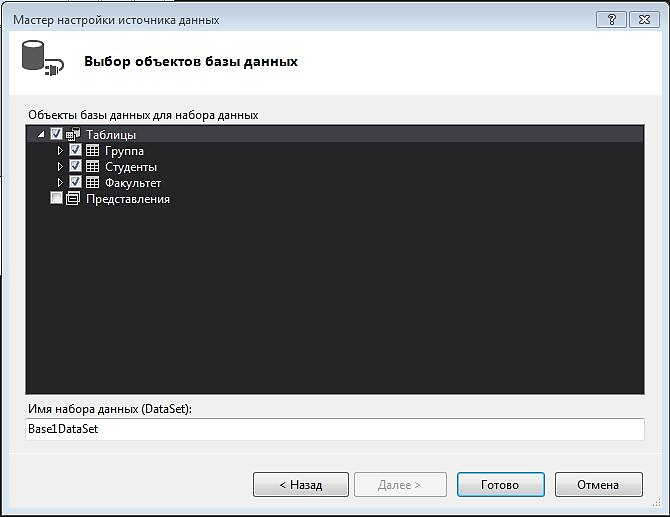


Рисунок 15. Мастер настройки источника данных

В этом окне требуется выбрать таблицы из БД, которые мы бы хотели видеть в своей программе. Нас интересуют соответственно все три. Отметив все поля, нажимаем «Готово» и возвращаемся вконструктор формы.

Теперь нужно снова нажать на выпадающее окно «Выберите источник данных» и выбрать появившеюся базу:

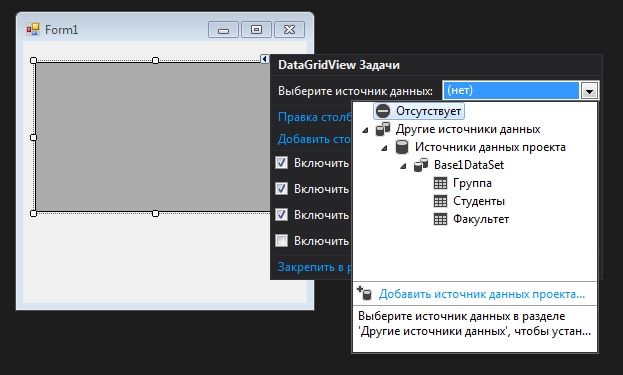


Рисунок 16. Выбор источника данных

Для первого элемента DataGridView выбираем **Факультет**.

Как мы видим на рисунке 17, в первом DataGridView появились заголовки столбцов таблицы факультетов, а на нижнюю панель добавились соответствующие этому DataGridView элементы BindingSource и TableAdapter.

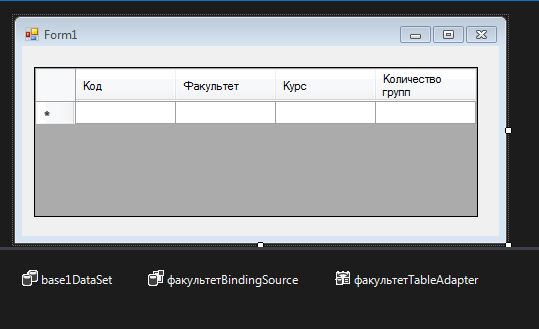
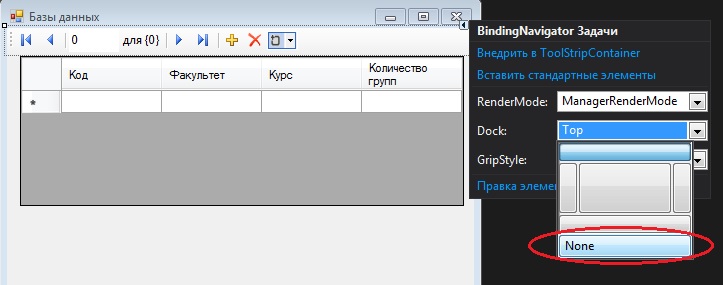


Рисунок 17. Отображение таблицы Факультет

Теперь добавим немного навигации. Помещаем на форму элемент управления **BindingNavigator**. При его перетаскивании на форму он появится вверху, поэтому нужно изменить у него свойство «Dock» на None, если хотите его переместить:

 Рисунок 18. Размещение на форме элемента **BindingNavigator**

Теперь нужно привязать BindingNavigator к DataGridView. Для этого открываем свойства элемента BindingNavigator и меняем пункт BindingSource:

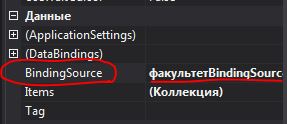


Рисунок 19. Свойства элемента **BindingNavigator**

Данный элемент позволяет просматривать и редактировать отображение в DataGridView, к которому он привязан. НО! Элемент ничего не изменяет в самой базе данных.

Теперь, аналогично, добавляем еще 2 элемента DataGridView и 2 элемента BindingNavigator, делая их по образу и подобию первых DataGridView и BindingNavigator, но с одним отличием для DataGridView. Источник данных для второго и третьего DataGridView будем выбирать иначе. В таблице выбора источника появился уже созданный нами при привязке таблицы Факультеты BindingSource, причем видно, что у него существуют подпункты, открываем эти подпункты и видим там подтаблицу отношения ФакультетГруппа, выбираем ее и получаем во второй DataGridView таблицу, которая отображает результаты для выбранного в первой таблице факультета.

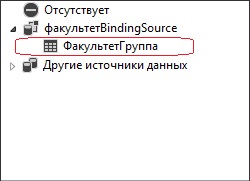


Рисунок 20. Выбор источника данных для отображения таблицы «Группа»

Аналогично для третьего DataGridView, выбираем уже созданный в результате предыдущего действия BindingSource и в нем - подтаблицу отношения ГруппаСтуденты.

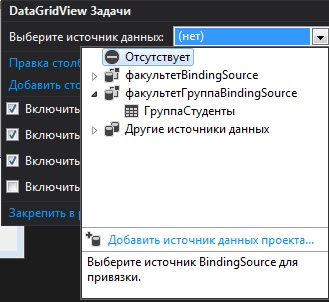


Рисунок 21. Выбор источника данных для отображения таблицы «Студенты»

Таким образом, мы получили все нужные нам таблицы с сохраненными между ними связями.

**Шаг 3. Реализация поиска и сортировки.**

Добавляем на форму два textBox и одну кнопку. Кнопка будет служить для сортировки, первый textbox (textbox1) для поиска по символам, а второй textBox (textbox2) для поиска целиком.

Прописываемкоддлясортировки:

privatevoidbutton1\_Click(object**sender**, EventArgs**e**)

{

if (dataGridView3.SortOrder==SortOrder.Ascending) dataGridView3.Sort(dataGridView3.Columns[1], ListSortDirection.Descending);

elsedataGridView3.Sort(dataGridView3.Columns[1], ListSortDirection.Ascending);

}

Код для поиска по символам:

privatevoidtextBox1\_TextChanged(object**sender**, EventArgs**e**)

{

for (int**i**=0; **i**<dataGridView3.RowCount-1; **i**++)

{

string**str**=dataGridView3.Rows[**i**].Cells[1].Value.ToString();

if (**str**.Contains(textBox1.Text) ==true) dataGridView3.Rows[**i**].Selected=true;

elsedataGridView3.Rows[**i**].Selected=false;

if (textBox1.Text=="") dataGridView3.Rows[**i**].Selected=false;

}

}

Коддляпоискацеликом:

privatevoidtextBox2\_TextChanged(object**sender**, EventArgs**e**)

{

for (int**i**=0; **i**<dataGridView3.RowCount-1; **i**++)

{

string**str**=dataGridView3.Rows[**i**].Cells[1].Value.ToString();

if (**str**==textBox2.Text) dataGridView3.Rows[**i**].Selected=true;

elsedataGridView3.Rows[**i**].Selected=false;

}

}

Результаты работы Приложения показаны на рисунках 22 – 26.

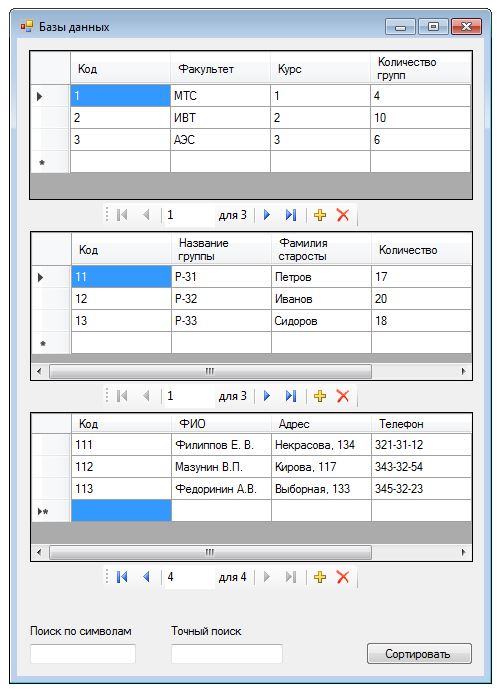


Рисунок 22. Вид формы сразу после запуска Приложения

|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\Никита\Desktop\БД\Результат 4.JPG | | C:\Users\Никита\Desktop\БД\Результат 5.JPG | |
| Рисунок 23.Сортировка (первое нажатие на кнопку) | | Рисунок 24.Сортировка (второе нажатие на кнопку) |
| C:\Users\Никита\Desktop\БД\Результат 1.JPG | | C:\Users\Никита\Desktop\БД\Результат 2.JPG |
| Рисунок 25.Поиск по наибольшему совпадению | | Рисунок 26.Точный поиск |