**ЛАБОРАТОРНАЯ РАБОТА № 7. «Обновление данных в формах»**

**Цель:** научиться организовывать обмен данными между формой и базой данных

**Теоретические сведения**

Обновление данных, содержащихся в формах и объектах.

Необходимость в обновлении данных в формах и объектах может возникать когда данные редактируются в одной форме, а используются в другой форме, например в качестве списка строк в ComboBox или в ListBox. После редактирования данных их необходимо сохранить и обновить те объекты, в которых эти данные используются.

Наиболее часто используемый сценарий такой процедуры следующий.

В основной таблице для заполнения некоторых полей используются ListBox или ComboBox.

Значения строк этих объектов хранятся в соответствующих таблицах той же базы данных (так называемые таблицы-справочники) и для редактирования этих таблиц созданы соответствующий формы. Если пользователь при заполнении основной таблицы обнаружил в ListBox или ComboBox ошибку в строке или отсутствие нужной строки он должен тут же перейти к редактированию соответствующего справочника (если он имеет на это право). Значит можно рядом с объектом ComboBox или ListBox разместить кнопку открывающую форму таблицы справочника. После окончания редактирования справочника и выхода и возврата в основную форму необходимо обновить содержимое соответствующего ComboBox или ListBox.

Так как возврат в основную форму происходит по определенной команде (например, закрытие формы-справочника), то целесообразно команду на обновление содержимого объектов разместить в обработчике события подачи данной команды.

В нашем примере ComboBox, строками которого являются строки таблицы «Должности», используется в форме «Сотрудники». Поэтому разместим кнопку открытия справочника «Должности» рядом с этим ComboBox и сделаем на ней надпись «Список должностей», как показано на рис. 1.

Переходы из одной формы в другую могут понадобиться, например для перехода из формы «Сотрудники» в форму «Должности» для редактирования справочника должностей в случае изменения названия должности или введения новой должности. Для перехода может быть использована та же процедура, что и при обычном открытии формы, а для того чтобы пользователь не мог вернуться в основную форму без закрытия справочника можно для его открытия использовать метод ShowDialog.

Для обновления comboBox после обновления данных в базе данных можно использовать следующий оператор

***positionsTableAdapter.Fill(kafedraDBDataSet.positions);***

Переходы из одной формы в другую могут понадобиться, например для перехода из формы «Сотрудники» в форму «Должности» для редактирования справочника должностей в случае изменения названия должности или введения новой должности. Для перехода может быть использована та же процедура, что и при обычном открытии формы, а для того чтобы пользователь не мог вернуться в основную форму без закрытия справочника можно для его открытия использовать метод ShowDialog.

Итак, таблица-справочник может быть открыта непосредственно из главной формы или из формы «Сотрудники». В первом случае при редактировании данных они должны быть просто сохранены в базе данных. При открытии формы «Должности» из формы «Сотрудники» в случае редактирования данных они должны быть сохранены в базе данных и должно быть произведено обновление списка строк объекта управления comboBox1. Таким образом режим сохранения данных (перед закрытием формы) должен зависеть от того, из какой формы была открыта форма «Должности». Эту информацию можно получить, проверив в форме «Должности» наличие открытой формы «Сотрудники». Если имеется открытая форма «Сотрудники», то необходимо сохранить данные в базе данных и затем обновить поле «comboBox1» в этой форме, в противном случае нужно только сохранить данные в базе данных. Проверить наличие открытой формы «Сотрудники» можно следующим образом.

* 1. При открытии формы «Должности» назначить ее владельцем (Owner) форму «Сотрудники».

Для этого обработчик события «нажатие кнопки Список должностей» может иметь вид

***FormPos f2 = new FormPos();***

***f2.Owner = this;***

***f2.ShowDialog();***

* 1. В форме «Должности» можно поместить две кнопки и надпись «Обновить». На одной из кнопок сделать надпись «НЕТ» (не обновлять). В обработчик события «Нажатие кнопки НЕТ» можно поместить единственный оператор закрытия формы без обновления

***this.Close();***

Навторой кнопке можно сделать надпись «ДА» и в обработчик нажатия этой кнопки поместить следующий код

***private void btYES\_Click(object sender, EventArgs e)***

***{***

***positionsBindingSource.EndEdit();***

***tableAdapterManager.UpdateAll(kafedraDBDataSet); // Обновление данных в БД***

***Form2 main = this.Owner as Form2;***

***if (main != null) //Если открыта форма 2 (Сотрудники). То обновляем comboBox1***

***{***

***int selInd = main.comboBox1.SelectedIndex; //запоминаем текущий индекс comboBox1***

***main.positionsTableAdapter.Fill(main.kafedraDBDataSet1.positions); //обновляем данные***

***main.comboBox1.SelectedIndex = selInd; //восстанавливаем исходный список***

***}***

***this.Close();***

***}***

После внесения вышеуказанных изменений работа с приложением будет выглядеть следующим образом.

При откоытии формы «Сотрудники» откроется окно, показанное на рис.1.

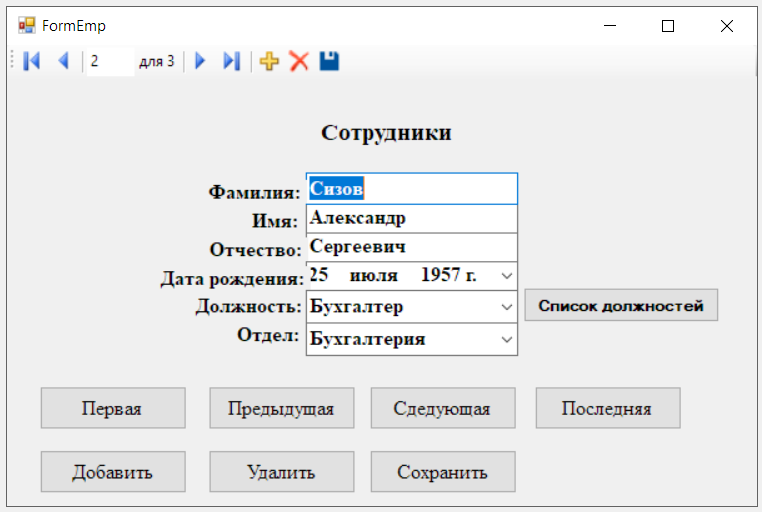


Рисунок 1. Внешний вид формы «Сотрудники»

В этом окне пользователь может ввести все данные о сотруднике и сохранить их, нажав копку «сохранить». В случае, если в списке должностей не окажется нужной должности, то пользователь может нажать кнопку «Список должностей» при этом откроется форма «Должности», показанная на рис. 2.

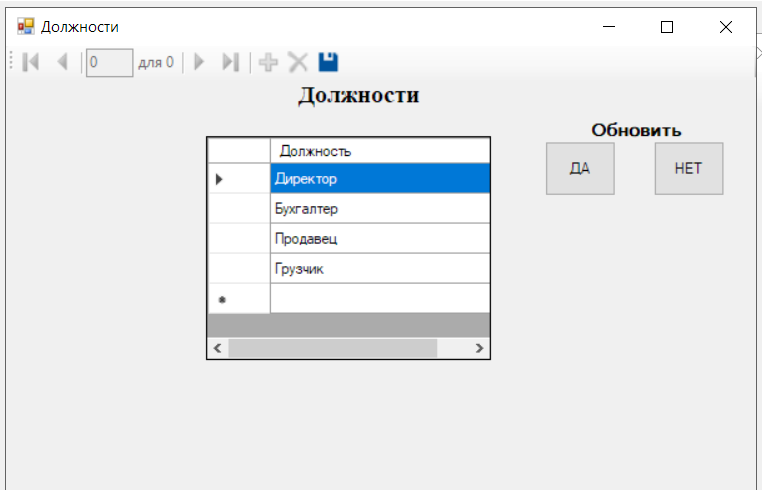


Рисунок 2. Внешний вид формы «Должности»

После обновлении списка должностей пользователь может нажать кнопку «ДА» для сохранения сделанных обновлений. Окно формы «Должности» закроется, а информация в форме «Сотрудники» - список должностей будет обновлена.

Рассмотрим случай обновления данных в пределах одной формы. Предположим, что в таблице «Сотрудники» необходимо сохранять информацию не только о должности сотрудника, но и об отделе, в котором он работает. В разных отделах могут быть как одинаковые, так и разные должности. И для каждой должности в таблице «Должности» предусмотрено поле dept\_id – идентификатор отдела. В этом случае нет необходимости вставлять в таблицу «Сотрудники» идентификатор отдела, так как идентификатор должности связан с идентификатором отдела. Но для того, чтобы пользователю упростить выбор должности сначал предусмотрим выбор отдела. При выборе отдела в поле «Сотрудники остаются только те должности, которые имеются в выбранном отделе.

Для этого в базе данных произведем следующие изменения.

Создадим таблицу «Отделы» (depts) с полями dept\_id и dept.

В таблицу «Должности» (positions) добавим внешний ключ dept\_id

В схеме базы данных установим связь между таблицами «Должности» и «Отделы» по ключам dept\_id

Схема базы данных примет вид, показанный на Рис. 3

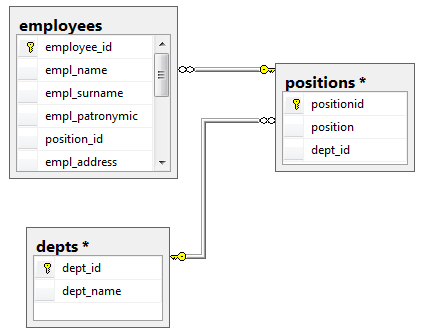


Рис.3. Схема базы данных

В приложении сделаем следующие изменения

В форму «Сотрудники» введем раскрывающийся список с названиями отделов (не связанный с данными в форме).

В форму «Должности» введем текстовое поле «Отдел» в котором будем отображать выбранный отдел, а в самой форме будем отображать только должности из выбранного отдела и кнопку список отделов для открытия формы редактирования списка отделов.

Добавим в проект форму «Отделы» для редактирования списка отделов.

В списке должностей формы «Сотрудники» будем отображать должности только для выбранного отдела.

Для выполнения всех изменений необходимо выполнить следующие действия

В форме «Сотрудники» изменить набор данных myShopDataSet. Для это нужно щелкнуть правой кнопкой мыши по объекту myShopDataSet и выбрать пункт меню «Изменить набор данных в конструкторе» . Откроется окно конструктора набора данных, показанное на рис. 4.

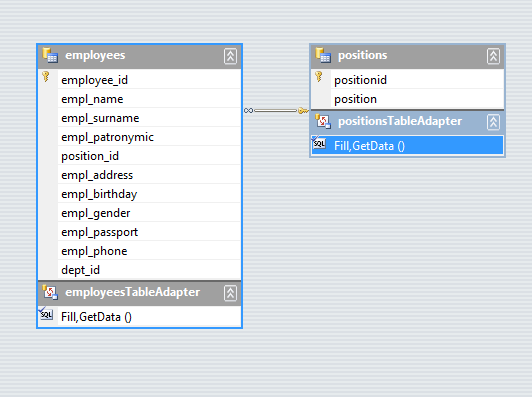


Рис.4 Окно конструктора набора данных

Щелкнуть правой кнопкой мыши по таблице positions («Должности») и выбрать в меню «Добавить» - «Столбец». После этого переименовать добавленный столбец в dept\_id.

Щелкнуть правой кнопкой мыши в поле конструктора и выбрать в меню «Добавить» - «TableAdapter». Откроется окно мастера подключения. В этом окне не изменяя подключения выбрать в форме «Выбор типа команды» - «Использовать инструкции SQL и ввести инструкцию SQL для выбора всех записей из таблицы «Отделы» (depts.) – SELECT \* FROM depts., последовательно нажать кнопки «Далее» и «Готово». Таблица «Отделы» (depts) появится в поле конструктора.

Связать таблицы «Должности» (positions) и «Отделы» (depts) по ключу dept\_id.

Открыть меню «Источники данных» и перетащить мышкой таблицу depts (Отделы) на форму 2 (Сотрудники). В форме 2 должен появиться объект deptsTableAdapter.

Установить свойство этого объекта Modifiers - Public

Поле со списком «Отделы» comboBox2 настроить аналогично полю «Должности» только без указания Выбранного значения. В этом данные из этого поля никуда вставляться не будут.

В форму «Должности» нужно добавить метку Отдел, свойство Text которой заполнять текстом из поля Отдел формы «Сотрудники»

Для этого в форму 3 может быть вставлен обработчик события загрузки формы следующего вида

***private void Form3\_Load(object sender, EventArgs e)***

***{***

***this.positionsTableAdapter.Fill(this.kafedraDBDataSet.positions);***

***// TODO: данная строка кода позволяет загрузить данные в таблицу***

***// "kafedraDBDataSet.employees". При необходимости она может быть***

***// перемещена или удалена.***

***Form2 main = this.Owner as Form2;***

***this.label4.Text = main.comboBox2.Text;***

***}***

Так как добавление новой строки может быть произведено как с помощью навигатора, так и самим пользователем, то внесение номера отдела удобнее сделать с помощью обработчика события щелчок мышкой по ячейке DataGridView следующего вида

***private void positionsDataGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)***

***{ Form2 main = this.Owner as Form2;***

***int id= (int) main.comboBox2.SelectedValue;//определение кода отдела***

***int n = positionsDataGridView.CurrentCell.RowIndex; //определение номера строки***

***this.positionsDataGridView.Rows[n].Cells[2].Value = id;//вставка кода отдела***

***}***

Для того, чтобы при выборе отдела в форме «Сотрудники» изменялся список должностей в поле «Должности» можно в обработчик события «изменение выбранного значения» поля «Отдел» вставить код изменения фильтра positionsBindingSource следующего вида

***private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)***

***{***

***this.positionsBindingSource.Filter="dept\_id=" + comboBox2.SelectedValue;***

***}***

Все вышеприведенные действия приводят к созданию приложения, в котором пользователь, не имеющий представления о базах данных может успешно работать с базой данных «Сотрудники»

**Задание на лабораторную работу**

В соответствии с индивидуальным заданием создать:

1. Кнопки для сохранения данных в базе данных в каждой форме для отображения таблиц-справочников (таблиц с первичными ключами).

2. В главной форме предусмотреть открытие таблиц-справочников и обновление соответствующих списков главной форме при закрытии форм таблиц-справочников.

**Содержание отчета**

1. Титульный лист

2. Тема и цель лабораторной работы

3. Задание для выполнения

4. Внешний вид и программный код созданных форм

5. Выводы

**Контрольные вопросы**

1. Как сохранить данные при закрытии фрмы?

2. Как обновить данные в главной из формы таблицы-справочника?

3.Как открыть новую форму?

4. Как отобразить полную информацию записи при выборе одного из полей в списке?