

ВВОД-ВЫВОД ДАННЫХ

Методические указания к лабораторной работе

1. ЦЕЛЬ РАБОТЫ

Целью работы является приобретение практических навыков проектирования и программирования пользовательских интерфейсов ввода-вывода данных СУБД с использованием технологии окон данных.

2. КРАТКАЯ ТЕОРЕТИЧЕСКАЯ СПРАВКА

Операции ввода-вывода данных являются одними из основных операций в СУБД. Для их выполнения организуются интерфейсы ввода-вывода данных, которые разделяются на *программные* (API) и *пользовательские* интерфейсы.

Данная работа посвящена разработке пользовательских интерфейсов ввода-вывода при помощи технологии окон данных.

2.1. ИНТЕРФЕЙСЫ ВВОДА-ВЫВОДА ДАННЫХ

В настоящее время самой распространённым типом пользовательского интерфейса является *оконный интерфейс*.

Оконный интерфейс представляет собой размещенные в окнах *элементы ввода данных* и *органы управления* ими, а также вспомогательные пассивные элементы: надписи, рисунки и т.п. Интерфейсы ввода-вывода могут также иметь элементы *мультимедиа*, но это характерно для тех СУБД, где используются *мультимедиа - данные*.

На рисунке 1 показан пример окна с элементами ввода данных и органами управления. На рисунке показаны:

- 1- строка ввода данных (Single LineEdit);
- 2- выпадающий список (DropDownListBox);
- 3- элемент выбора (CheckBox)
- 4- контроллер с инициализированным окном данных (DataWindow control);
- 5- кнопка (CommandButton);
- 6- надпись (StaticText).

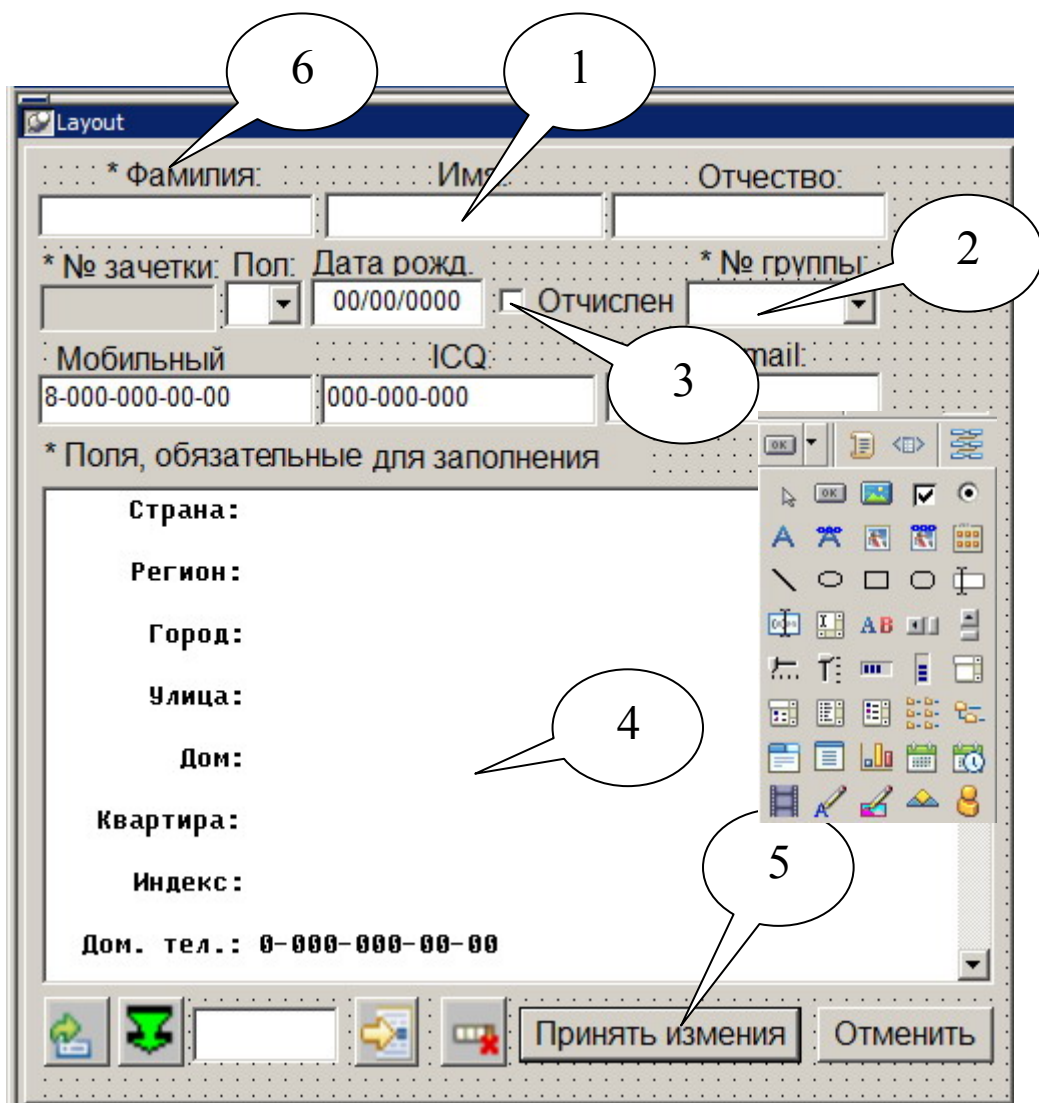


Рис. 1. Пример окна с элементами ввода данных и органами управления.

Показанные на рис. 1 элементы другие подобные элементы управления интерфейсом можно выбрать в соответствующей палитре в мастерской окон системы PowerBuilder, показанной также на рис.1.

2.2. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ОКОН ДАННЫХ.

Поскольку в технологии окон данных с объектом окна данных связан шаблон оператора SELECT SQL, использование окон данных для вывода данных является вполне естественным. Однако данная технология применяется не только для вывода, но и для ввода данных, как показано в лабораторной работе «Технология окон данных».

В данной работе рассматривается применение технологии окон данных в задачах ввода-вывода данных для нескольких связанных таблиц базы данных. Связанные таблицы могут иметь различное назначение, определяемое характером использования их данных. Рассмотрим

соответствующие варианты организации интерфейсов для нескольких связанных таблиц.

2.2.1. Иерархически связанные таблицы и выпадающие окна данных.

Иерархически связанные таблицы образуют отношения «родитель – потомок». Часто родительская таблица является справочной, данные которой используются в других таблицах. В таблицах – справочниках имеется первичный ключ и их собственные поля. В них нет внешних ключей, первичные ключи таких таблиц представлены как внешние ключи в других, дочерних таблицах.

На рис. 2 показан пример иерархически связанных таблиц.

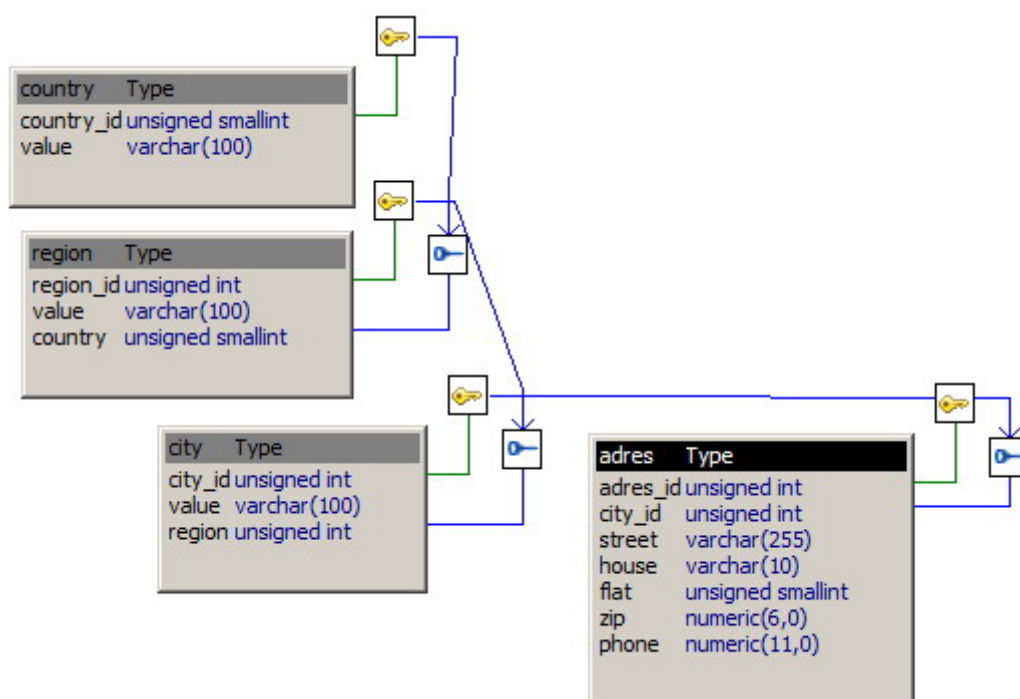


Рис. 2. Пример иерархически связанных таблиц.

Иерархия таблиц на рис. 2 начинается с таблицы стран, далее идут таблицы областей и городов. Коды городов в качестве внешних ключей находятся в таблице адресов. Соответственно, заполнять таблицы на рис. 2 данными следует начиная с таблицы стран. Она может быть заполнена и далее редактируема при помощи интерфейса, основанного на использовании окна данных, построенного для одной таблицы. Такой интерфейс рассмотрен в лабораторной работе «Технология окон данных».

Рассмотрим интерфейс ввода-вывода данных для двух иерархически связанных таблиц на примере таблиц стран и областей, показанных на рис.2.

Первичный ключ в таблице стран желательно построить автоинкрементным, чтобы не вводить его значения. Для таблицы районов этого делать не

следует, поскольку, например, для России удобно считать ключом номер региона, используемый ГИБДД.

Далее строится окно данных, которое использует только одну таблицу районов с выбранными полями, как показано на рис. 3.

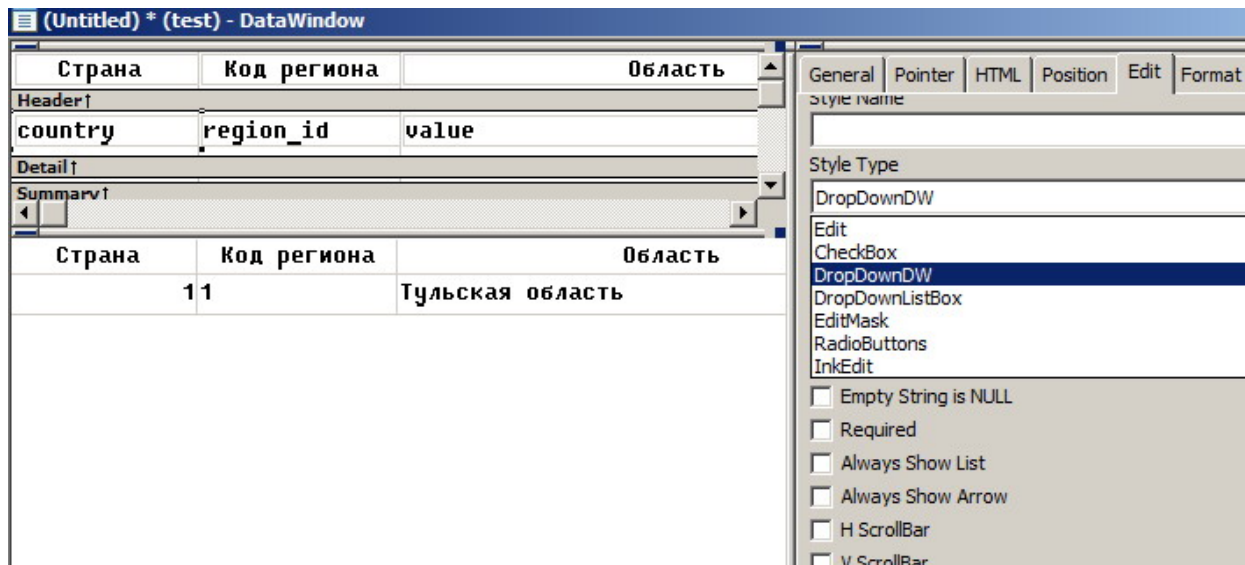


Рис. 3. Окно данных для таблицы районов на рис. 2.

В закладке **Edit** в окне свойств выбранного поля **country** окна данных в выпадающем списке **Style Type** необходимо выбрать стиль **DropDownDW**. Это означает применение т.н. выпадающего окна данных в поле **country**, в котором мы будем выбирать страну, для которой задаем область.

Выбрав этот стиль, необходимо его настроить:

- указать то окно данных, которое будет использовано для ввода данных в поле (поле Data Window);
- указать поле выпадающего окна данных, данные которого будут показаны (поле Display Column);
- указать поле выпадающего окна данных, данные которого будут введены в поле **country** (поле Data Column)

На рис. 4 показан пример настройки выпадающего окна данных. Дополнительно на рис. 4 выбраны:

- применение горизонтальной прокрутки (**H Scroll Bar**) в выпадающем окне данных;
- автоматическая инициализация выпадающего окна данных в поле **country** (**Auto Retrieve**);
- ширина выпадающего окна данных, совпадающая с шириной поля **country** (**Width of Drop Down**).

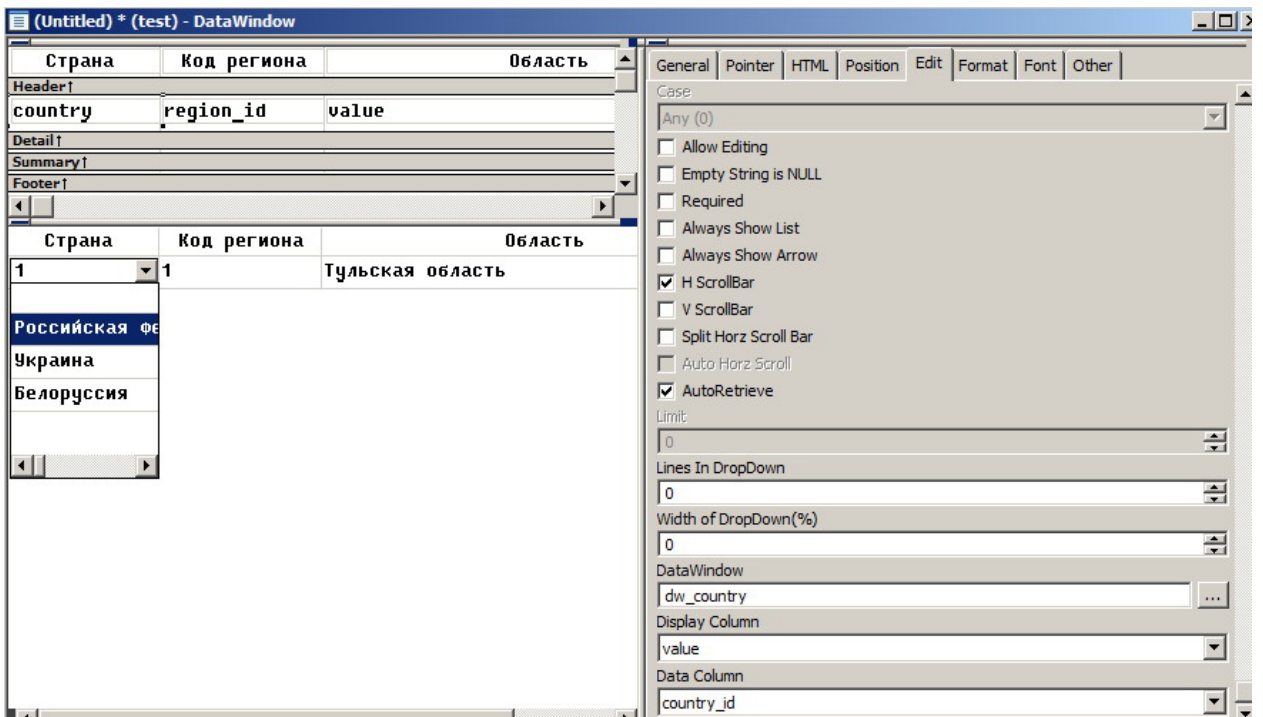


Рис. 4. Настройки параметров выпадающего окна данных

Данное решение позволяет, используя в основном окне данных одну таблицу районов, вводить в нее коды стран, выбирая их по названиям стран из другой таблицы стран, представленной в выпадающем окне данных.

2.2.2. Окна данных для нескольких таблиц.

В общем случае в интерфейсах ввода-вывода необходимо вводить данные «одновременно» в несколько связанных таблиц. В этом случае также можно использовать технологию окон данных.

Рассмотрим пример связанных таблиц, показанных на рис. 5.

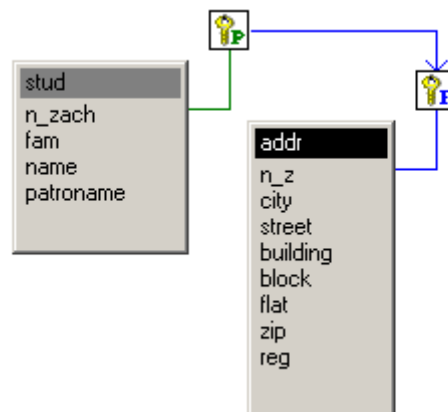


Рис. 5. Таблицы, связанные не иерархично

Поскольку все таблицы на рис. 5 равноправны, данные в них должны вводиться синхронно. Рассмотрим, как это можно сделать в технологии окон данных.

Построим окно данных, макет источника данных для которого показан на рис. 6.

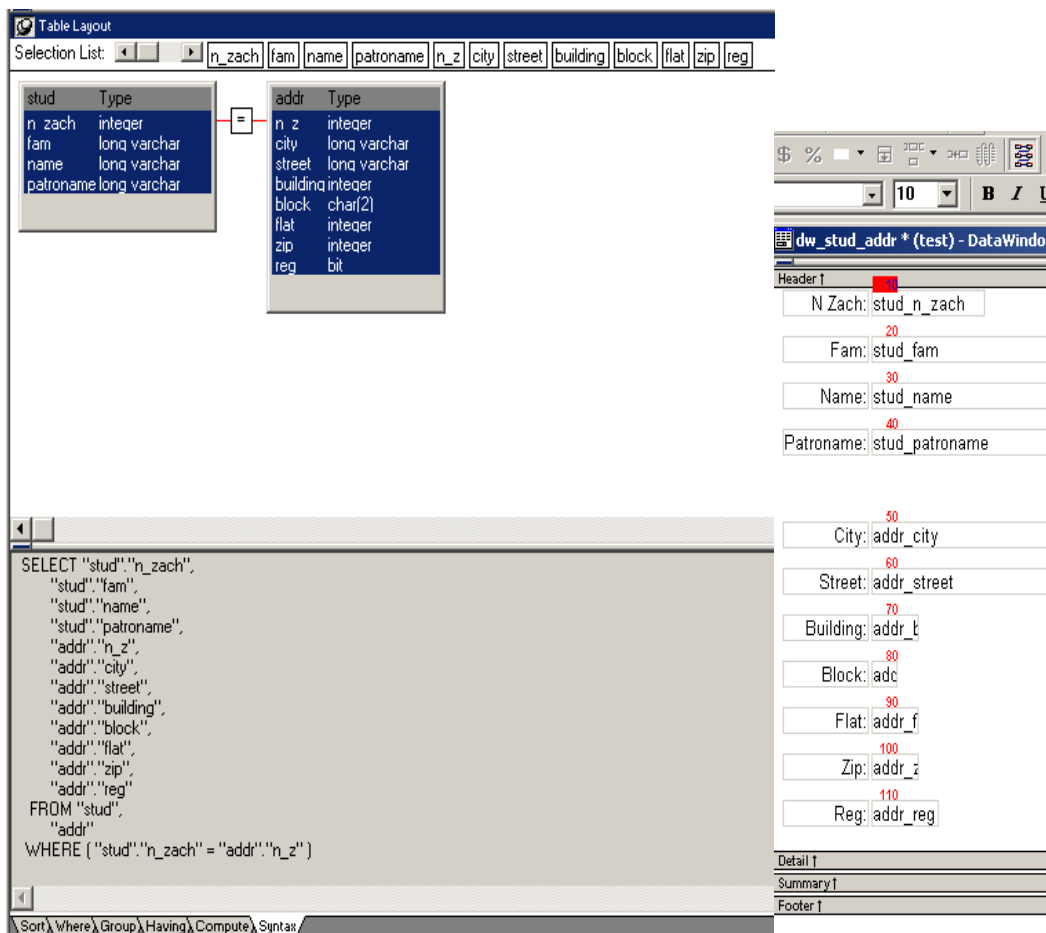



Рис. 6. Макет источника данных для окна данных таблиц на рис. 5.

Назначим номера, определяющие порядок выбора всех полей всех таблиц в окне данных. Для этого нажмем клавишу  системного меню и расставим номера у всех полей, как показано на рис. 6. После этого станет возможным редактирование всех полей в окне данных.

В технологии окон данных всякое редактирование полей окна данных должно завершаться обновлением таблиц базы данных, представленных в окне.

Если для обновления применить известный скрипт `dw_1.Update()`, то обновятся поля только одной таблицы – той, для которой установлены параметры обновления. Проверить параметры обновления таблиц можно, вызвав окно задания параметров обновления (**Specify Update Properties**), как показано на рис. 7, 8. Из рис. 8 видно, что в выпадающем списке **Table to Update** можно выбрать только одну таблицу – в данном случае таблицу `stud`.

Следовательно, для ввода данных в несколько таблиц необходимо организовать переключение таблиц и задание их параметров обновления. Это

можно сделать не только посредством окна задания параметров обновления, но и программно.

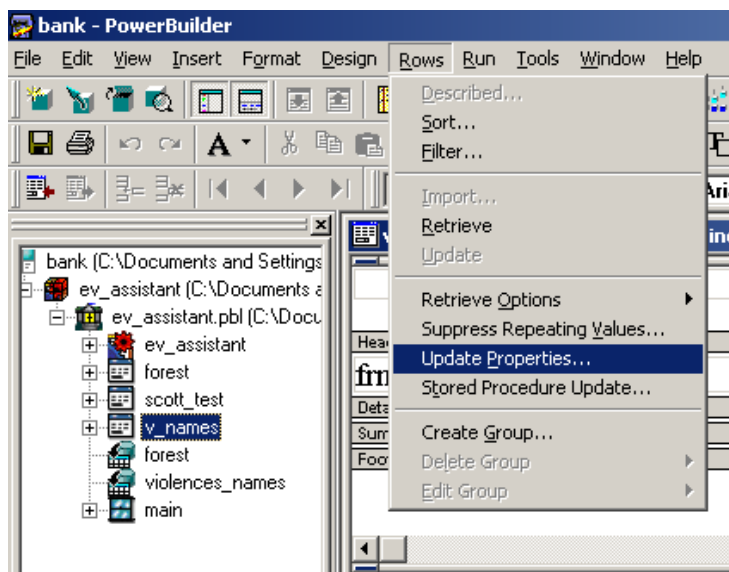


Рис. 7. Вызов окна задания параметров обновления таблиц.

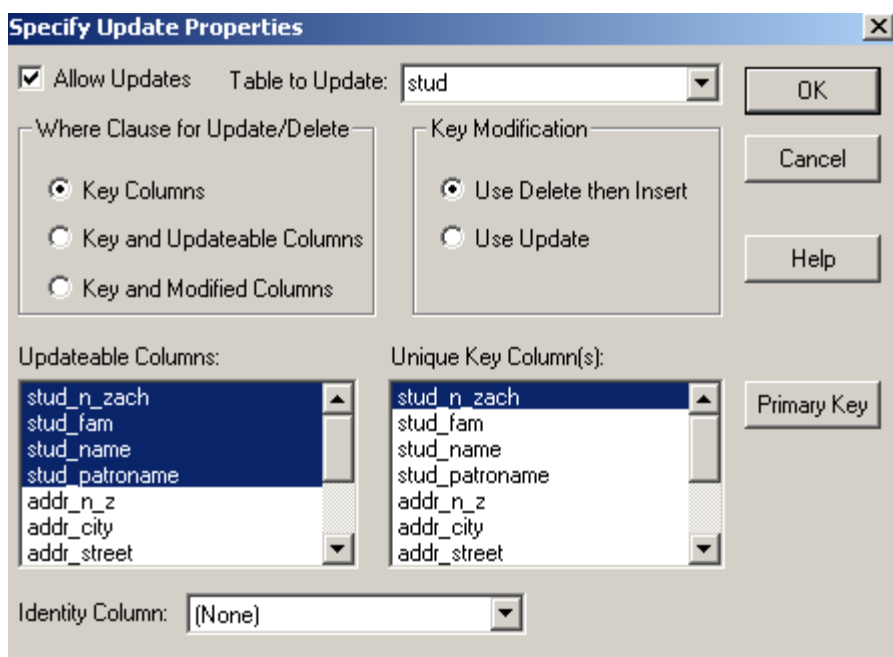


Рис. 8. Окно задания параметров обновления таблиц.

Прежде чем показать программное решение для обновления нескольких таблиц, поясним смысл органов управления окна задания параметров обновления.

Allow Updates - элемент выбора. Его необходимо выбрать, если предполагается обновление данных в таблице через объект DataWindow. По умолчанию такие обновления разрешены, если окно данных содержит поля только одной таблицы и в них входят ключевые поля.

Table to Update – выпадающий список. Если окно данных содержит поля нескольких таблиц, то необходимо указать имя таблицы, в которой будут обновления через данное окно данных.

Where Clause for Update/Delete - группа радио кнопок. Определяет, какие поля включаются в опцию **WHERE** операторов **Update/Delete**.

Key Modification - группа радио кнопок. Определяет, какие генерируются операторы SQL, когда поле, указанное в окне **Unique Key Column(s)** изменено.

Use DELETE then INSERT используется, если в СУБД не разрешено пользователям модифицировать два ключа или устанавливать какое – либо значение поля в одной записи, равное значению поля в другой записи, или если СУБД имеет триггер **Insert**.

Use UPDATE используется, если в СУБД только одна запись может быть изменена перед обновлением БД.

Updateable Columns – окно. В данном окне можно выбрать и подсветить обновляемые поля. Если такие поля выбраны, то их порядковые номера табуляции отличны от нуля – тогда их может выбрать пользователь при работе приложения.

Unique Key Column(s) – окно. Здесь указываются поля, которые идентифицируют обновляемую запись. По умолчанию считаются поля первичного ключа.

Primary Key- кнопка. Выбирается, чтобы вместо окна **Unique Key Column(s)** использовать первичные ключи.

Identity Column – поле ввода. Это особое поле, по которому сервер идентифицирует записи в таблице.

Таким образом, если окно данных содержит поля только одной таблицы и среди них есть ключевое поле, то они автоматически задаются как обновляемые поля с ненулевыми значениями номеров, определяющих порядок выбора таких полей. Пользователи могут менять данные в таких полях непосредственно в окне данных.

Если окно данных содержит поля нескольких таблиц или вида, то они автоматически задаются как необновляемые поля с нулевыми значениями

номеров, определяющих порядок выбора таких полей. Пользователи не могут менять данные в таких полях в окне данных;

Рассмотрим программное решение для обновления нескольких таблиц в окне данных. Ниже приведен скрипт, обычно помещаемый в кнопку обновления в окне, предназначенном для ввода данных в несколько таблиц.

Пример 1

```
long cr
cr=dw_1.GEtROw()
boolean lb_ok
string is_updateerror
integer li_return

lb_ok = True // пока не обнаружено ни одной ошибки
is_updateerror = "" //пока нет сообщений об ошибке

li_return = dw_1.Update(True,False) //Update в таблице stud

IF li_return <> 1 OR is_updateerror <> "" THEN
lb_ok = False
END IF
IF lb_ok THEN
  dw_1.object.DataWindow.Table.UpdateTable = "addr"
  dw_1.object.addr_city.Update = "yes"
  dw_1.object.addr_street.Update = "yes"
  dw_1.object.addr_building.Update = "yes"
  dw_1.object.addr_block.Update = "yes"
  dw_1.object.addr_flat.Update = "yes"
  dw_1.object.addr_zip.Update = "yes"
  dw_1.object.addr_reg.Update = "yes"
  dw_1.object.addr_n_z.Update = "yes"

  dw_1.object.stud_n_zach.Key = "no"
  dw_1.object.stud_n_zach.Update = "no"
  dw_1.object.stud_fam.Update = "no"
  dw_1.object.stud_name.Update = "no"
  dw_1.object.stud_patroname.Update = "no"

  dw_1.object.addr_n_z[cr] = dw_1.object.stud_n_zach[cr] // здесь номер строки
  аргумент

li_return = dw_1.Update(True,False) //Update в таблице addr

IF li_return <> 1 OR is_updateerror <> "" THEN
lb_ok = False
END IF

// Возвращаем в исходное
dw_1.object.DataWindow.Table.UpdateTable = "stud"
dw_1.object.addr_city.Update = "no"
dw_1.object.addr_street.Update = "no"
dw_1.object.addr_building.Update = "no"
dw_1.object.addr_block.Update = "no"
```

```
dw_1.object.addr_flat.Update = "no"  
dw_1.object.addr_zip.Update = "no"  
dw_1.object.addr_reg.Update = "no"  
dw_1.object.addr_n_z.Update = "no"  
dw_1.object.stud_n_zach.Key = "yes"  
dw_1.object.stud_n_zach.Update = "yes"  
dw_1.object.stud_fam.Update = "yes"  
dw_1.object.stud_name.Update = "yes"  
dw_1.object.stud_patroname.Update = "yes"
```

```
END IF
```

```
IF lb_ok THEN  
COMMIT;  
dw_1.ResetUpdate()  
ELSE  
ROLLBACK;
```

```
MessageBox("Error - Обновление БД не прошло", &  
is_updateerror, StopSign!)  
END IF
```

3. ОБОРУДОВАНИЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАБОТЫ

Работа выполняется на персональном компьютере с операционной системой Windows. В качестве инструментальной СУБД используется SAP-Sybase PowerBuilder версии не ниже 8-й.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Изучить краткую теоретическую справку.

4.2. Используя технологию окон данных, построить интерфейс ввода-вывода для таблиц, показанных на рис. 2. Интерфейс должен содержать окно с кнопками добавления, удаления и сохранения данных.

4.3. Используя технологию окон данных, построить интерфейс ввода-вывода для таблиц, показанных на рис. 5. Интерфейс должен содержать окно с кнопками добавления, удаления и сохранения данных. Применить скрипт из примера 1.

4.4. Ответить на контрольные вопросы и выполнить упражнения.

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет по работе должен содержать:

- Макеты окон данных для выбранных таблиц;

- Макеты окон интерфейсов ввода-вывода
- Тексты скриптов, реализующих сценарии управления событиями.

При защите работы студент должен продемонстрировать умение применять технологию окон данных для создания интерфейсов ввода-вывода для таблиц, указанных преподавателем.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ И УПРАЖНЕНИЯ

1. Какие характерные объекты принадлежат технологии DataWindow?
2. Что такое SQLCA?
3. Почему по умолчанию поля нескольких таблиц в окне данных установлены как необновляемые?
4. Как обеспечивается ссылочная целостность базы данных в СУБД?
5. **Упражнение 1.** В интерфейсе ввода-вывода для нескольких таблиц проверить корректность удаления строк в окне данных при используемом в кнопке сохранения скрипте из примера 1. Модифицировать программное решение для обеспечения корректности удаления.
6. **Упражнение 2.** В базе данных atm.db для таблиц, выбранных преподавателем, создать интерфейс ввода-вывода.
7. **Упражнение 3.** Если в поле окна данных применяется выпадающее окно данных, то отображаются все его поля, хотя из него вводятся данные только одного поля. Как сделать так, чтобы отображались только нужные поля выпадающего окна данных?

7. Литература

1. Электронный ресурс: <http://lis.tula.ru/>
2. Богатырев М.Ю. Разработка и программирование систем управления базами данных. - Тула, изд-во ТулГУ, 2009. - 145 с.
3. Богатырев М.Ю. Введение в систему Power Builder. Методические указания к выполнению лабораторных работ. - Тула, изд-во ТулГУ, 1998. - 36 с.
4. Смит Б. Дж., Шаад Г.У. Power Builder 5.0. Библия разработчика. - К.: Диалектика, 1997. - 544 с.
5. Хайес В.Б. Использование Power Builder 6. - Киев: Вильямс, 1998. - 688 с.