

ТЕХНОЛОГИЯ ОКОН ДАННЫХ (DATA WINDOW)

Методические указания к лабораторной работе

1. ЦЕЛЬ РАБОТЫ

Целью работы является приобретение практических навыков создания и использования окон данных в приложениях, работающих с базами данных.

2. КРАТКАЯ ТЕОРЕТИЧЕСКАЯ СПРАВКА

Технология окон данных предназначена для организации ввода – вывода и поиска данных в СУБД, работающих в архитектуре «клиент – сервер».

2.1. ПРИНЦИПЫ ТЕХНОЛОГИИ ОКОН ДАННЫХ

В СУБД сервер базы данных постоянно получает множество запросов от клиентов, которые он должен обработать и выполнить за минимальное время. Предположим, что оператор - клиент СУБД вводит данные в таблицы базы данных, пользуясь, например, клавиатурой компьютера. Если разрешить ему непосредственно вводить данные в нужные таблицы, то, независимо от скорости работы оператора, время, затраченное на ввод данных, будет казаться вечностью по сравнению с теми интервалами времени (измеряемые миллисекундами), которые требуются серверу на обработку и выполнение запросов. В таком режиме ввода данных сервер СУБД будет простаивать, а если подобных клиентов оказывается несколько, то производительность СУБД упадет до нуля.

Необходимо решение, исключаящее подобные задержки, и такое решение реализовано в технологии окон данных. Суть решения заключается в следующем.

1. Клиент получает в пользование данные тех таблиц, с которыми ему необходимо работать, путем однократного запроса к серверу.
2. Клиент автономно, не обращаясь к серверу, выполняет с полученными данными все необходимые ему операции изменения, удаления и вставки данных.
3. Закончив работу с данными, клиент извещает сервер о необходимости обновления базы данных в соответствии с результатами его операций с

данными предоставленных ему таблиц. Сервер обновляет базу данных, соблюдая при этом принцип целостности.

Очевидно, что данный способ работы клиента с данными более эффективен, чем описанный ранее, поскольку клиент не загружает сервер постоянно, а лишь дважды: запросив данные на редактирование, и запросив сервер обновить базу данных.

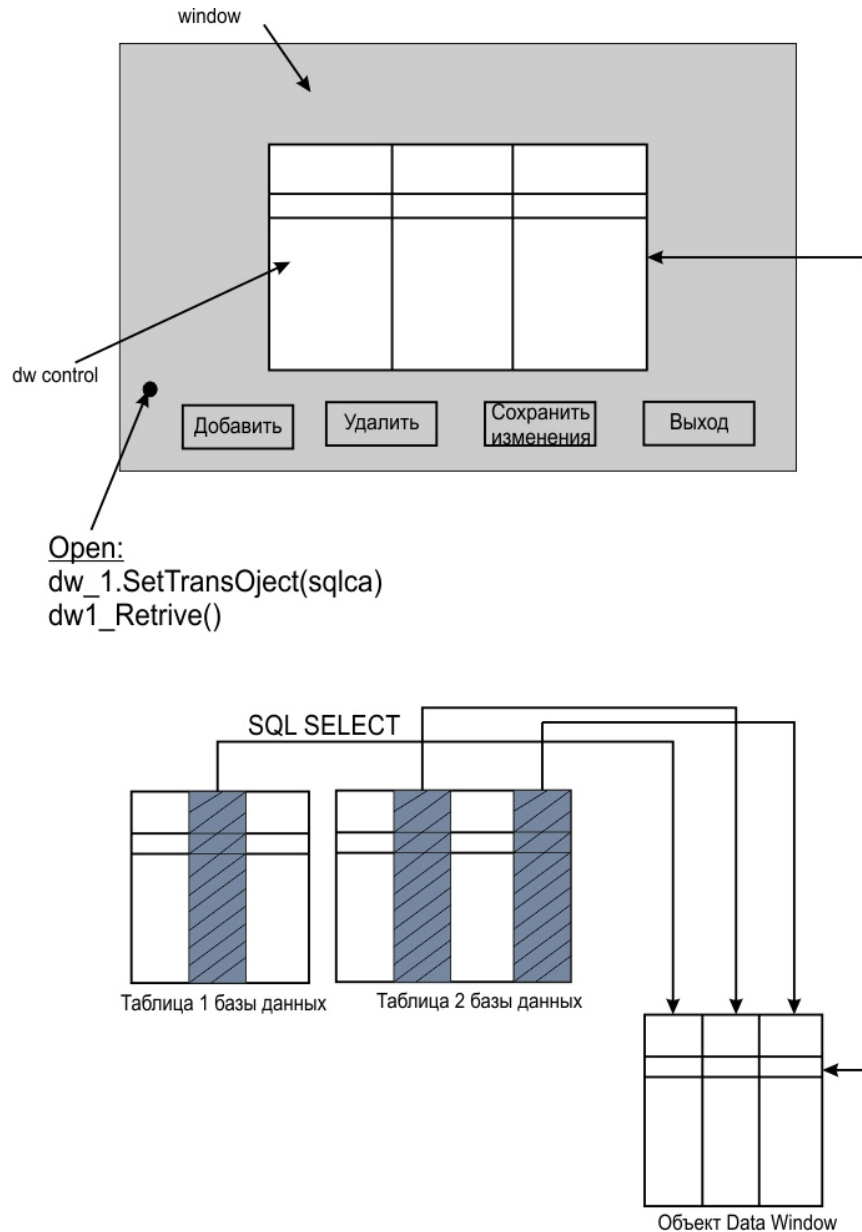


Рис. 1. Объекты, применяемые в технологии окон данных

Рассмотрим, как реализуются принципы 1-3 технологии окон данных, пользуясь рисунком 1.

В технологии окон данных применяются:

- объект «Окно данных» (DataWindow);

- стандартное окно, в котором отображаются и редактируются данные (Window);
- контроллер окна данных (DataWindow Control), предназначенный для управления данными в окне данных.

Окно данных (DataWindow) представляет собой шаблон оператора SELECT языка SQL и форму представления данных. Оператор SELECT необходим для извлечения данных тех таблиц, с которыми необходимо работать (см первый принцип). Шаблон оператора SELECT задает структуру оператора, в которую в качестве параметров помещаются имена таблиц, имена полей и условия связывания таблиц. Извлеченные при помощи оператора SELECT данные могут быть показаны несколькими способами – это определяется заданием формы представления данных.

Стандартное окно. Отображение извлечённых данных и управление им выполняется в стандартном окне (Window) при помощи контроллера окна данных (DataWindow Control) и органов управления – например, кнопок, показанных на рис. 1.

Контроллер окна данных (DataWindow Control) управляет объектом «Окно данных» (DataWindow) при помощи функций, вызываемых из других объектов, например, кнопок. Так при инициализации кнопки «Добавить» на рис. 1 вызывается функция контроллера, вставляющая пустую строку в окно данных.

Показанная на рис. 1 схема взаимодействия объектов технологии окон данных имеет ряд преимуществ.

2.1.1. Разделение интерфейса и данных.

Один и тот же интерфейс, например, показанное на рис. 1 окно с кнопками и контроллером, может применяться для управления данными различных таблиц.

Какими таблицами управлять, определяется в объекте «Окно данных» (DataWindow), а как управлять – функциями контроллера, вызываемыми посредством кнопок. Контроллеру окна данных можно задавать разные объекты DataWindow – как на этапе проектирования интерфейса, так и при его работе, то есть программно. Сам интерфейс при этом остается неизменным. Это придает большую гибкость программным решениям в среде Sybase PowerBuilder.

Используя окна данных, можно реализовывать различные решения для интерфейсов ввода – вывода данных:

- *окна запроса,*

- форма для ввода данных,
- отчеты.

Далее мы познакомимся со всеми этими решениями.

2.1.2. Как устроены окна данных

Технология окон данных позволяет, применяя один и тот же интерфейс, организовать ввод, редактирование и поиск данных, отображая результаты этих операций. Рассмотрим, как это делается.

Когда приложение выбирает строки из базы данных в окно Data Window, система сохраняет эти строки в *буферах*. Обычно буферы находятся в ОЗУ компьютера - клиента.

Существует четыре буфера окон данных:

- первичный (Primary) буфер
- буфер фильтра (Filter)
- буфер удаления (Delete)
- буфер оригинала (Original)

В первичном буфере находятся строки, непосредственно находящиеся в окне данных.

Буфер фильтра содержит строки, выбранные с помощью команды SELECT, но затем отфильтрованные дополнительными условиями.

Буфер удаления содержит строки, которые сначала были в первичном буфере, но затем удалены с помощью функции DeleteRow.

Буфер оригинала содержит строки, которые были выбраны из БД функцией Retrieve(), до того, как пользователь начал изменять их.

Power Builder позволяет работать непосредственно с буферами окна данных. Значения каждой строки и каждого столбца первичного буфера образуют двумерный массив

data[r,c],

где r - номер строки, c - номер столбца. Обращаться к буферам можно непосредственно, указывая их имена в выражениях, например:

v=dw_1. object. a.filter[r]

```
v=dw_1. object. a.delete[r]  
v=dw_1. object. a.delete.original[r,c]
```

2.2. ПОСТРОЕНИЕ ОКОН ДАННЫХ

Для того, чтобы построить один из интерфейсов ввода – вывода, необходимо определить состав таблиц базы данных, с которыми необходимо работать и построить соответствующие объекты окон данных DataWindow. При выполнении приложения PowerBuilder окна данных всегда находятся в составе стандартного окна PowerBuilder.

Для создания объекта DataWindow требуется запустить мастер окон данных (DataWindow Painter).

Вызова мастера окон данных:

- нажать кнопку New панели инструментов,
- выбрать закладку DataWindow.

В открывающемся окне необходимо выбрать один из стилей представления (Presentation Style) данных в окне, как показано на рис. 2

2.2.1. Выбор стиля представления данных

Выбор стиля представления определяет, каким образом данные должны быть представлены в окне.

В системе PowerBuilder предусмотрены следующие стили представления, показанные на рис.2.

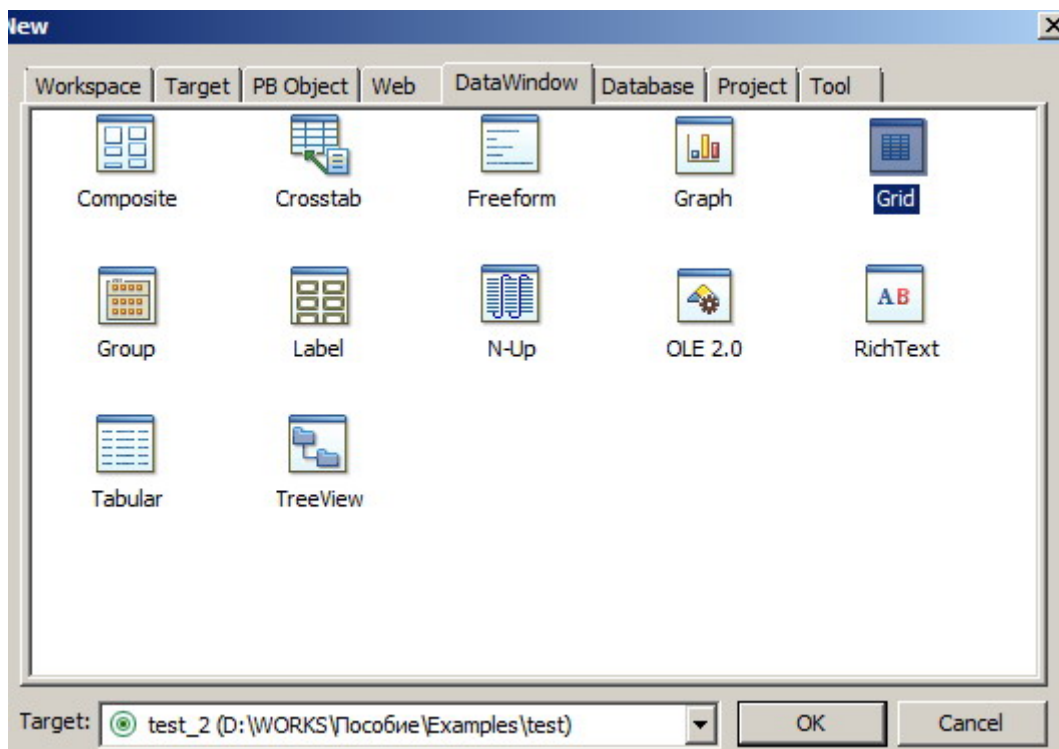


Рис. 2. Выбор стиля представления данных окна данных.

1. **Grid (Сетка).** Используя этот стиль, PowerBuilder помещает данные в матрицу с ячейками, что довольно удобно, когда необходимо работать с несколькими записями таблиц.
2. **FreeForm (Свободный стиль).** В этом стиле данные и подписи можно поместить в произвольном порядке, поэтому он широко используется для самых разнообразных целей, например, для создания формальных писем и конвертов.
3. **Tabular (Таблица).** Так же, как и в стиле Grid, данные в окне DataWindow представлены в виде таблиц. Этот стиль более гибкий по сравнению со стилем Grid и дает возможность значительно улучшить внешний вид данных. Стиль Tabular очень часто используется при оформлении отчетов. Кроме того, он хорош для ввода данных в окно DataWindow.
4. **Group (Группа).** Очень похож на стиль Tabular, но дополнительно позволяет автоматически добавлять заголовки, номера страниц, итоговые и промежуточные значения данных.
5. **Label (Почтовая наклейка).** Позволяет создавать почтовые наклейки.
6. **N-Up (N-колоночная разбивка).** Разновидность таблицы, используется для более компактного отображения данных на странице, располагая данные в две или несколько колонок.
7. **Graph (Диаграмма).** Позволяет представить числовые данные в графическом виде.
8. **CrossTab (Сводная таблица).** Позволяет отобразить разнообразные числовые данные в матрице, напоминающей электронную таблицу.
9. **Composite (Комбинированный стиль).** В этом стиле в одном окне можно расположить несколько окон данных DataWindows. Стиль Composite очень удобен для создания отчетов по управлению и его можно использовать для ввода и отображения данных в окне DataWindow.
10. **Rich Text (Текст в расширенном формате).** С помощью стиля Rich Text можно создавать письма и другие документы, заполняя шаблон форматированного главного окна DataWindow данными из базы. Этот стиль позволяет достаточно широко использовать возможности форматирования, ориентированные на обработку текстов, например, верхние и нижние колонтитулы, а также многочисленные шрифты.
11. **OLE 2.0.** (OLE – технология связывания и внедрения объектов). С помощью данного стиля можно помещать в окна данных OLE – объекты, в качестве которых могут выступать диаграммы, схемы, электронные таблицы и т.д.
12. **TreeView (Древовидный стиль).** Этот стиль позволяет изображать данные в виде графа – дерева. Удобен для отображения иерархических структур данных.

2.2.2. Выбор источника данных (Data Source).

Выбрав стиль представления, далее получаем окно, показанное на рис.

3

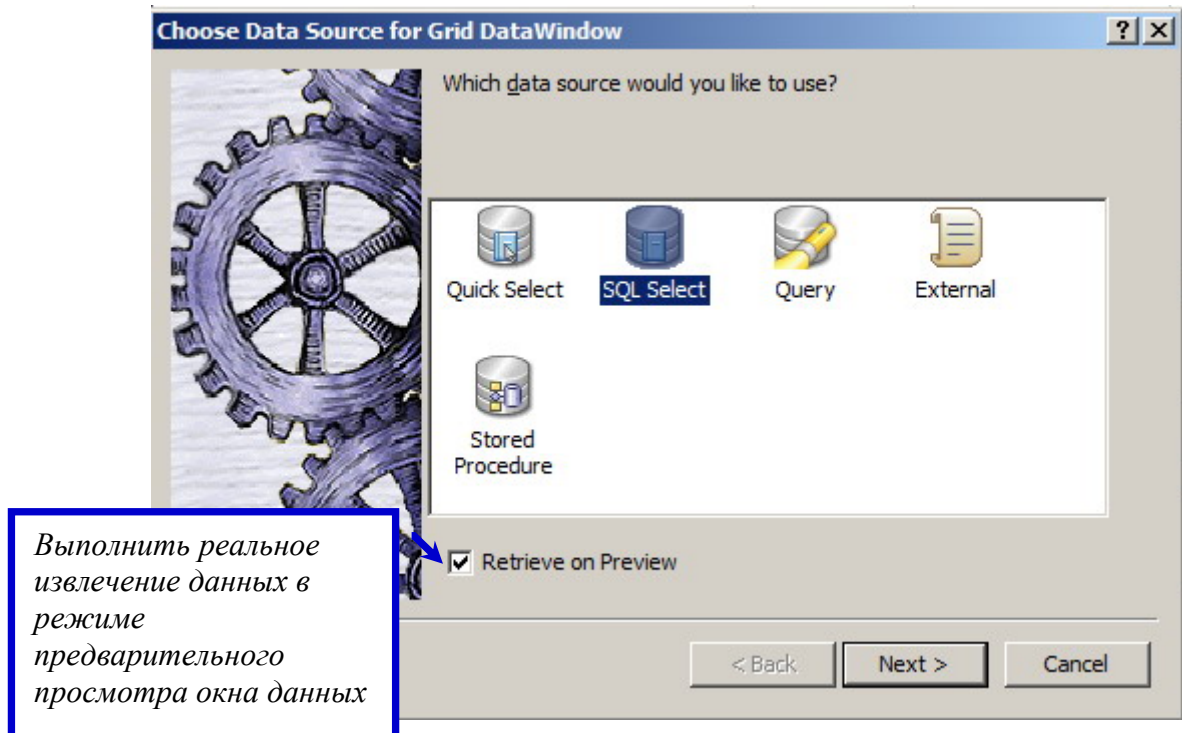
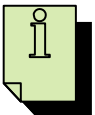


Рис. 3. Выбор источника данных для окна данных.

Источник данных определяет способ получения данных для окна.

Обратите внимание, что термин «источник данных» имеет два значения:

- источником данных для программы – приложения называется настройка драйвера ODBC, которая позволяет подключиться к базе данных конкретной платформы;
- источником данных для окна данных согласно рис. 3 является один из вариантов извлечения данных из таблиц или иных объектов базы данных, к которой имеется подключение.



Как видим, существует пять вариантов источников данных для окон данных. Первые три варианта используют запрос SQL SELECT, а остальные – нет. Рассмотрим их подробнее.

1. **Quick Select (Быстрый выбор).** Основываясь на сделанных разработчиком приложения настройках, PowerBuilder генерирует оператор SQL SELECT, который можно редактировать в мастере Select Painter.
2. **SQL Select (Выбор SQL-запроса).** Выбор пиктограммы SQL Select приведет непосредственно к вызову мастера Select Painter, где можно построить SQL-запрос до вызова мастера окон данных DataWindow Painter. Конечный результат при выборе как Quick

Select, так и SQL Select – одинаковый, а именно SQL-запрос SELECT.

3. **Query (Запрос).** В мастере Select Painter можно создать запрос как SQL-оператор SELECT с именем и сохранить его в библиотеке проекта. При выборе этого варианта PowerBuilder покажет список уже созданных запросов и предложит выбрать один из них. Когда выбирается нужный запрос, PowerBuilder копирует его и использует в качестве источника данных окна DataWindow. Находясь в мастере окон данных, можно также просматривать и изменять этот оператор с помощью Select Painter. Следовательно, как и в двух предыдущих случаях, конечным результатом является оператор SQL SELECT. Однако, здесь PowerBuilder не организует связь окна данных с запросом, а создает копию запроса SELECT. Следовательно, если изменится запрос, сделанные им изменения не будут отображены в окне DataWindow, и наоборот.
4. **External (Внешний источник данных).** Выбрав данный источник, далее необходимо задать структуру таблицы, в которую будут помещаться данные из какого-либо внешнего источника – например, из таблицы – текста.
5. **Stored Procedure (Хранимая процедура).** При выборе этого варианта источника данных DataWindow получает строки, вызывая хранимую в базе данных процедуру.

2.2.3. Построение оператора SELECT

На следующем этапе построения окон данных строится SQL - оператор SELECT, в результате выполнения которого в первичный буфер окна данных поступают данные базы.

Оператор строится с помощью конструктора Select Painter, который открывается после выбора источника данных.

В конструкторе Select Painter необходимо заполнить шаблон оператора SELECT:

- выбрать таблицы базы данных, с которыми будем работать;
- выбрать отображаемые поля таблиц;
- задать дополнительные ограничения.

На рис. показаны выбор таблиц в конструкторе Select Painter и созданный им оператор SELECT.

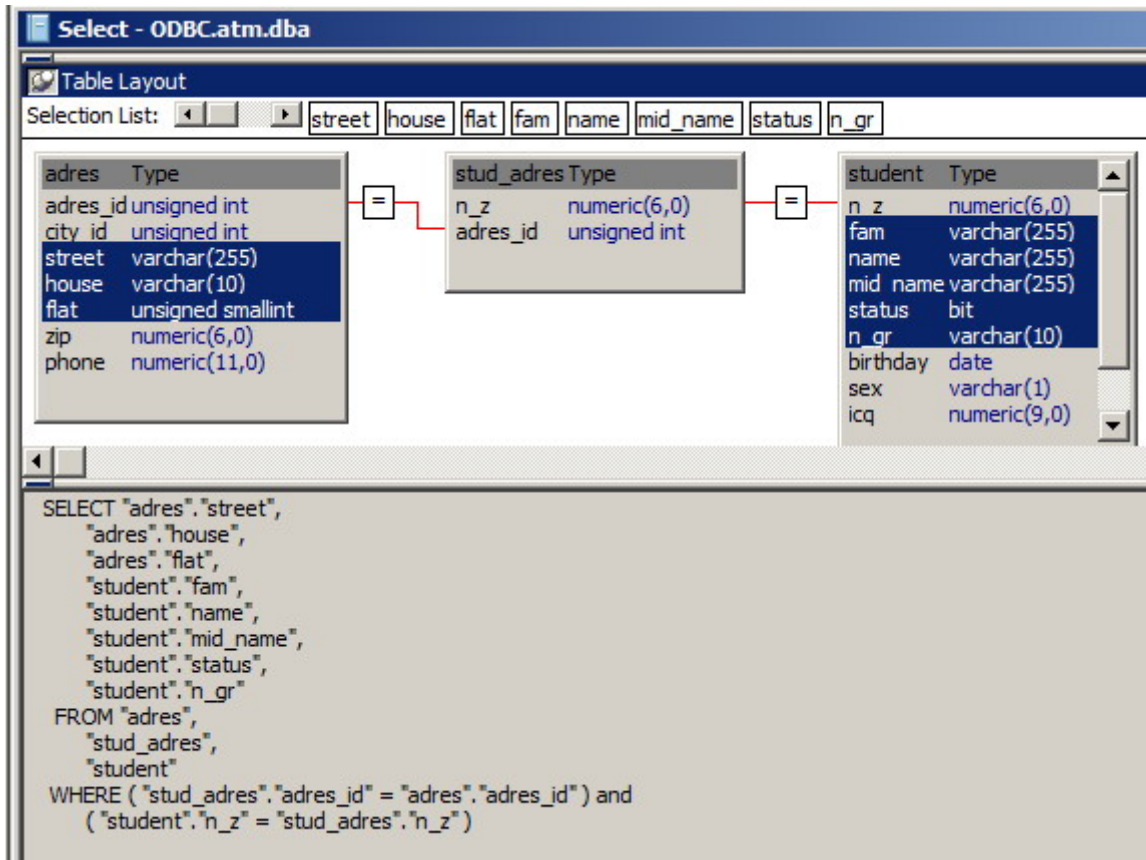
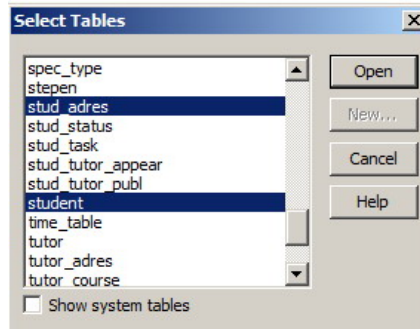


Рис. 4 Построение оператора SELECT окна данных

Отметим две особенности:

- при выборе таблиц в конструкторе Select Painter отображаются таблицы базы данных, к которой PowerBuilder подключен в соответствии с активным профилем подключения;
- Если выбраны связанные таблицы (как на рис.), то конструктор Select Painter автоматически формирует условия связи в опции WHERE оператора.



2.2.4. Построение макета окна данных

После построения оператора SELECT в конструкторе кнопка Return конструктора передает управление мастерской окон данных, где задается макет окна данных. Пример макета показан на рис. 5.

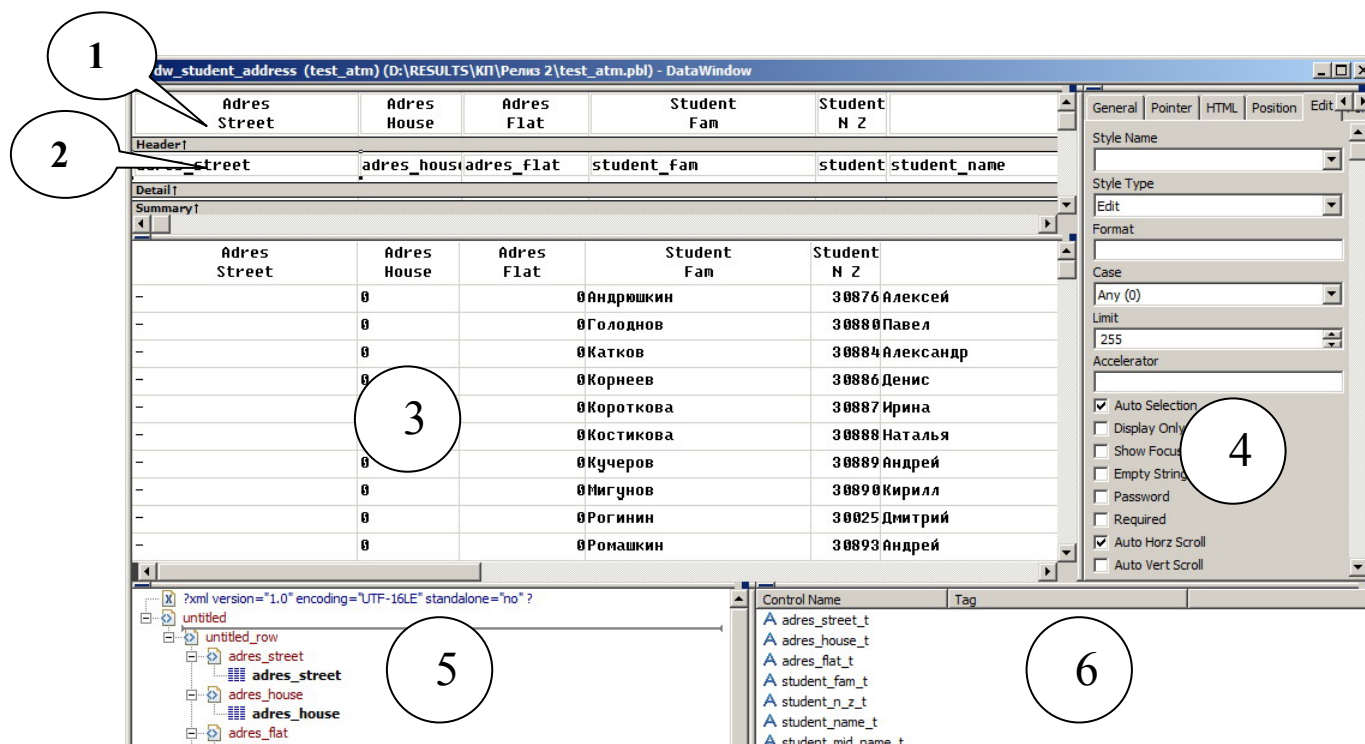


Рис. 5. Пример макета окна данных. 1 – заголовки полей, 2 – данные, 3 – результат извлечения данных, 4 – окно свойств выбранных объектов макета, 5 – XML – модель окна данных, 6 – список объектов макета.

На рис. 5 показан пример макета окна данных с выбранным стандартным представлением «сетка» (grid) в виде таблицы. Возможны другие 11 представлений, рассмотренные выше.

2.3. ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ОКНАМИ ДАННЫХ

Для управления окнами данных необходимо иметь три основных компонента, показанные на рис. 1:

1. объект - окно данных (DataWindow), создаваемый в мастерской окон данных (DataWindow Painter);

2. окно, разрабатываемое в мастерской окон (Window Painter), в котором располагаются все остальные объекты и элементы управления;

3. контроллер окна данных, размещаемый в окне и связанный с объектом DataWindow.

Окна создаются в мастере окон (Window Painter).

Для вызова мастера окон требуется нажать кнопку New панели инструментов и выбрать пиктограмму Window закладки PB Object, как показано на рисунке 5.

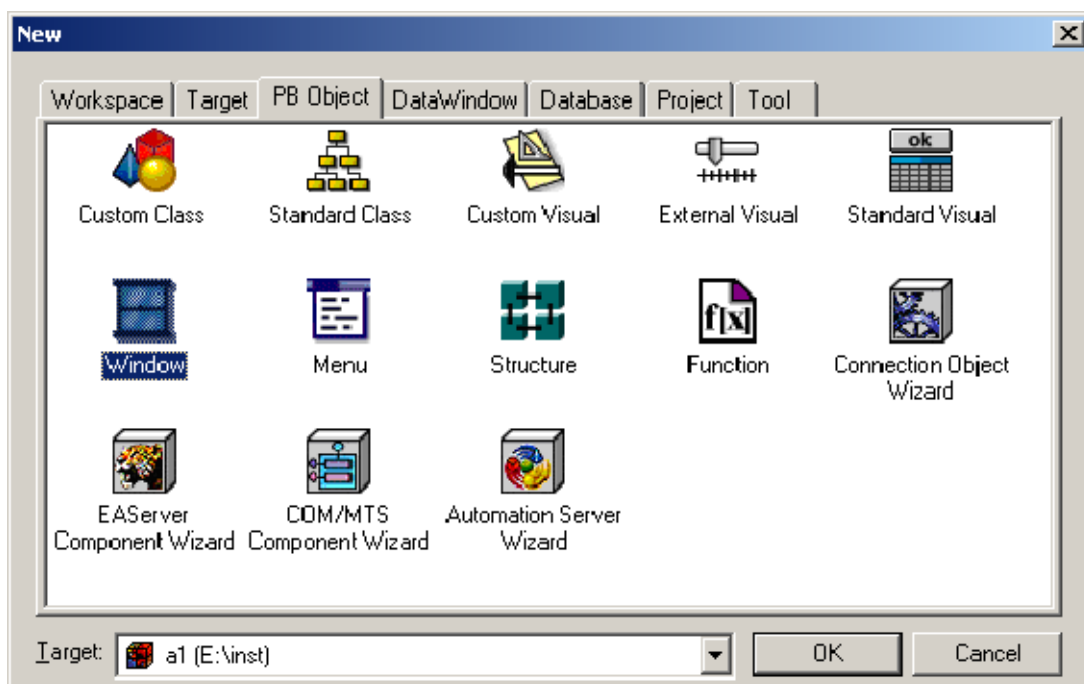


Рис. 5. Вызов мастера окон

В мастере окон следует создать главное (main) окно, в котором будут размещены элементы управления: поля ввода данных, кнопки, элементы выбора и т.д.

Размещение контроллера окна данных практически ничем не отличается от размещения в окне других элементов управления – см. рис. 6. Необходимо воспользоваться контекстной палитрой мастерской окон, показанной на рисунке. Кнопка **Create DataWindow control** предназначена для создания контроллера.

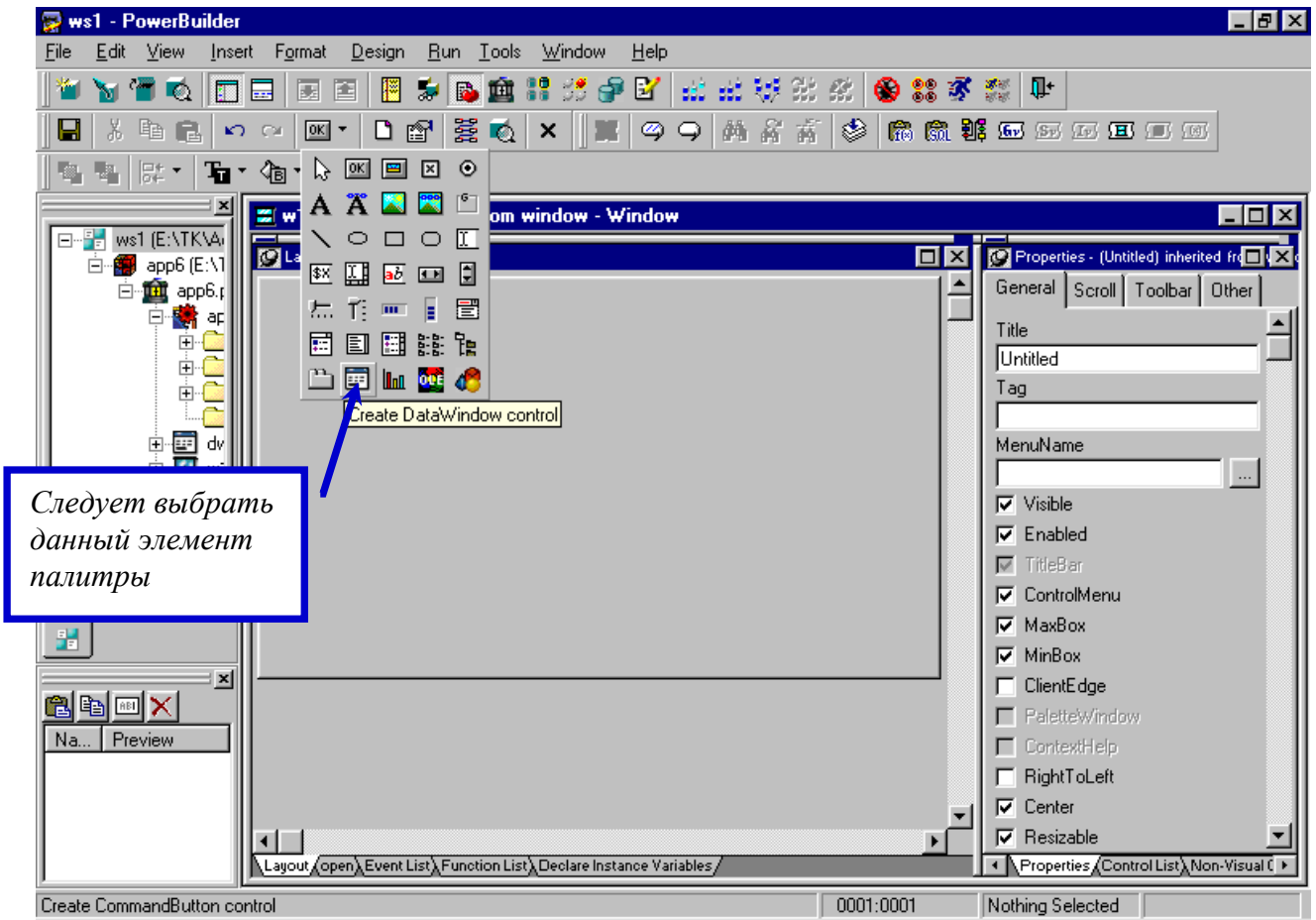


Рис. 6. Создание контроллера окна данных.

Для назначения объекта окна данных контроллеру требуется в окне свойств контроллера выбрать и назначить объект данных (DataObject), как показано на рис. 7.

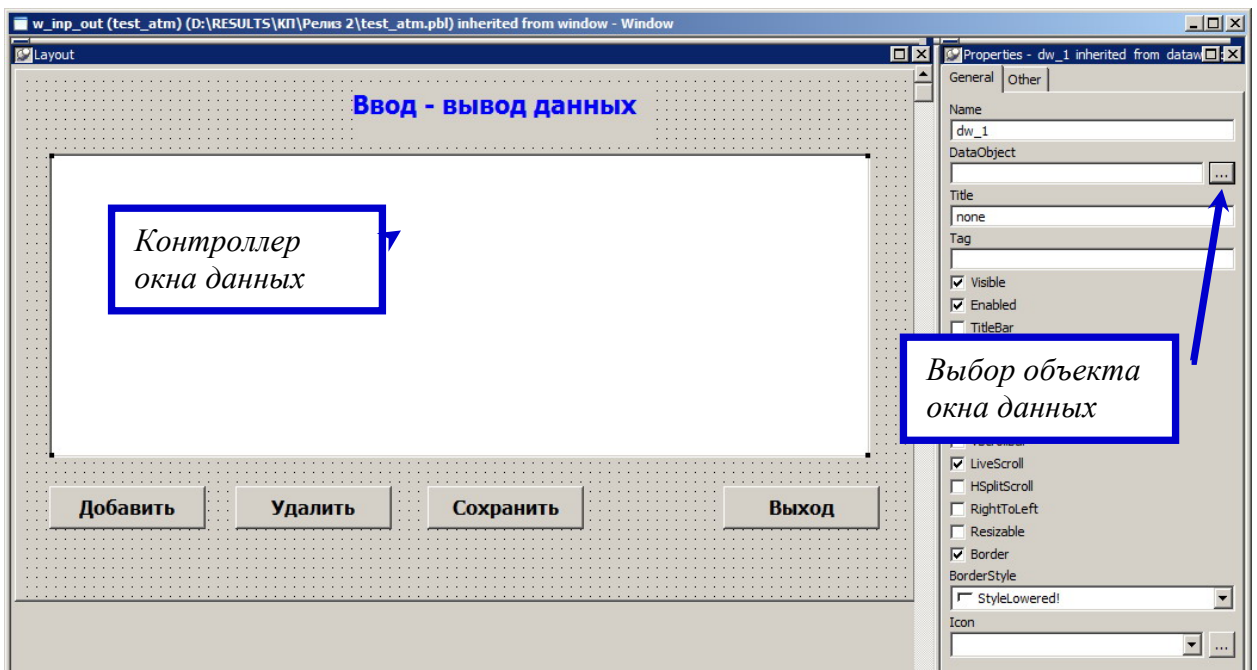


Рис. 7. Назначение объекта окна данных контроллеру.

Обратим внимание, что объект окна данных и контроллер окна данных имеют разные имена.

2.3.1. Управление событиями

В мастерской окна данных и в других мастерских, где возможно управление событиями, применяется язык высокого уровня PowerScript для программирования сценариев управления.

В системе PowerBuilder применяется «глубокая» поддержка принципов объектно-ориентированного программирования. Это выражается, в частности, в том, что программный код создается главным образом как реакция на события, связанные с объектами. Например, событие открытия окна или событие нажатие кнопки в окне должны быть обработаны по соответствующим сценариям, описываемым фрагментами кода (скриптами) на языке PowerScript.

В данном случае событию открытия окна следует назначить сценарий, который будет инициализировать окно данных, связанное с контроллером.

В качестве альтернативы можно использовать событие конструктор (constructor) контроллера окна данных.

Соответствующий скрипт на языке PowerScript имеет вид:

```
dw_1.SetTransObject(sqlca)
dw_1.Retrieve()
```

Здесь `dw_1` - имя контроллера, а после точки указываются функции контроллера. Функция `SetTransObject` назначает стандартный объект транзакции `sqlca`, который используется сервером при работе с базой данных.

Функция `Retrieve` возвращает в окно данных строки из таблицы. Если не вызвать эту функцию при открытии окна, то окно данных, отображаемое с помощью контроллера, останется пустым.

В окне данных можно производить изменения строк таблицы, но они не будут приняты базой данных, пока не будет выполнен оператор `UPDATE`. Контроллер окна данных имеет функцию `Update (x, y)`, которая задает параметры оператора `UPDATE` и инициирует его выполнение. Простейший вариант этой функции имеет вид:

```
dw_1.Update()
```

Эта функция будет обновлять в базе данных те строки, в которые пользователь внесет изменения, работая в окне данных, если в нем разрешено

производить такие изменения. Если макет окна данных построен для одной таблицы, то по умолчанию обновления строк окна данных разрешены.

Функцию `Update()` следует применить в скрипте, описывающем сценарий обработки события нажатия кнопки «Сохранить» в окне на рис. 7.

Приведем перечень функций, применяемых в сценариях обработки событий нажатия остальных кнопок в окне на рис. 7.

Кнопка «Добавить»: функция `InsertRow()`, кнопка «Удалить»: функция `DeleteRow()`. Применение этих функций может потребовать вызова других функций, необходимых для реализации конкретного сценария.

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 принципы технологии окон данных.

4.2. Используя технологию окон данных, построить интерфейс ввода-вывода для таблиц базы данных, созданной в первой лабораторной работе согласно выбранному варианту исходных сущностей.

4.3. ответить на контрольные вопросы и выполнить упражнения.

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет по работе должен содержать:

- Макеты окон данных для выбранных таблиц;
- 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. Какое из приведенных обращений к окну данных будет правильным?
 - 2.1.w_name.dw_1.Retrieve()
 - 2.2.dw_1.Retrieve()
3. Что означает выражение dw_1.object.stud_name ?
4. Для чего применяется технология окон данных?
5. Что такое SQLCA?
6. Без какой функции PowerScript невозможно выполнить извлечение данных?
7. В окне данных, созданном в среде PowerBuilder, по умолчанию можно ли обновлять данные в нескольких таблицах?
8. Какая функция выполняет извлечение строк из таблиц базы данных и отображение их в окне 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.

