Тема 23. Разработка многотабличной базы данных

Задание.

Построим базу данных библиотечной организации. Основными объектами БД являются Читатели, Писатели и Книги. Писатели – авторы книг, читатели – потребители книг. В общем случае, одна книга может иметь несколько авторов, а один писатель может быть автором нескольких книг. Читатель может держать на руках несколько разных книг, а одна книга может одновременно использоваться несколькими читателями (при условии, что в библиотеке имеется несколько экземпляров этой книги).

Общая схема связи объектов представлена на рис. 23.1.

Связь множеств **Читатели** и **Писатели** «типа многие ко многим» (M:M) представляется отношением **Абонемент**, каждый кортеж которого связывает некоторую книгу и одного из читателей этой книги. Связь множеств **Книги** и **Писатели** представляется отношением **Авторы** типа M:M, кортежи которого связывают книгу и одного из ее авторов.

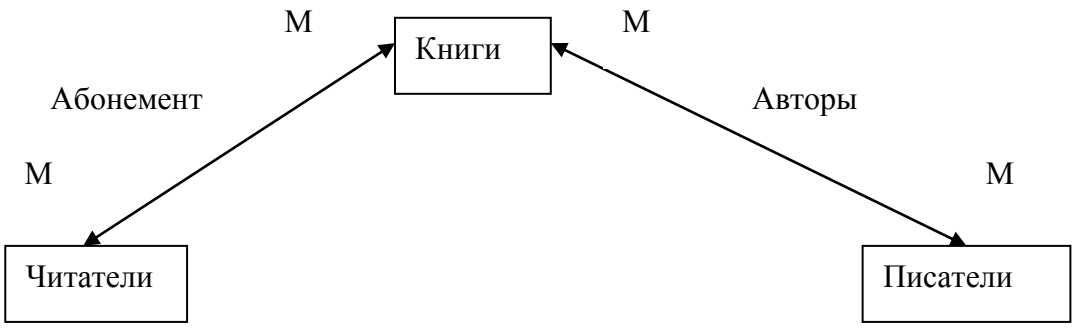


Рис. 23.1.Связи наборов объектов.

Наборы объектов **Читатели, Книги, Писатели** и связывающие их отношения **Абонемент** и **Авторы** в реляционной модели представляются соответствующими таблицами.

Схемы таблиц.

Таблица **Books** (книги).

Атрибуты:

- **BkId** – идентификатор книги, тип **Autoincrement**, ключ таблицы;

- **Name** – название книги, тип **Alpha** (30);
- **Number** – количество экземпляров, тип **Short**;
- **Note** – аннотация, тип **Memo** (30).

Вторичный индекс **BkInd** по полю **Name**.

Таблица **Writers** (писатели).

Атрибуты:

- **WrId** – идентификатор, тип **Autoincrement**, ключ;
- **Name** – имя писателя, тип **Alpha** (30);
- **Biography** – биография писателя, тип **Memo** (30).

Вторичный индекс **WrInd** по полю **Name**.

Таблица **Readers** (читатели).

Атрибуты:

- **RdId** – идентификатор, тип **Autoincrement**, ключ;
- **Name** – имя читателя, тип **Alpha** (20);
- **Idate** – дата записи, тип **Date**;
- **Note** – замечания, тип **Memo** (30).

Вторичный индекс **RdInd** по полю **Name**.

Таблица **Abonement** (абонемент).

Атрибуты:

- **RdId** – идетификатор читателя, тип **Long Integer**, ключевой атрибут;
- **BkId** – идентификатор книги, тип **Long Integer**, ключевой атрибут.

Таблица **Authors** (авторы).

Атрибуты:

- **BkId** – идентификатор книги, тип **Long Integer**, ключевой атрибут;
- **WrId** – идентификатор писателя, тип **Long Integer**, ключевой атрибут.

Задание 1.

С помощью программы **Database Desktop** создайте БД **Bibliotheca**, включающую таблицы **Books, Readers, Writers, Authors, Abonement** и индексы **BkInd, WrInd, RdInd**. Занесите несколько записей в каждую таблицу. Обратите внимание, что значения полей типа Memo не отображаются в окне **Database Desktop**, но все введенное сохраняется в базе данных.

Задание 2. Начало разработки приложения «Библиотека»

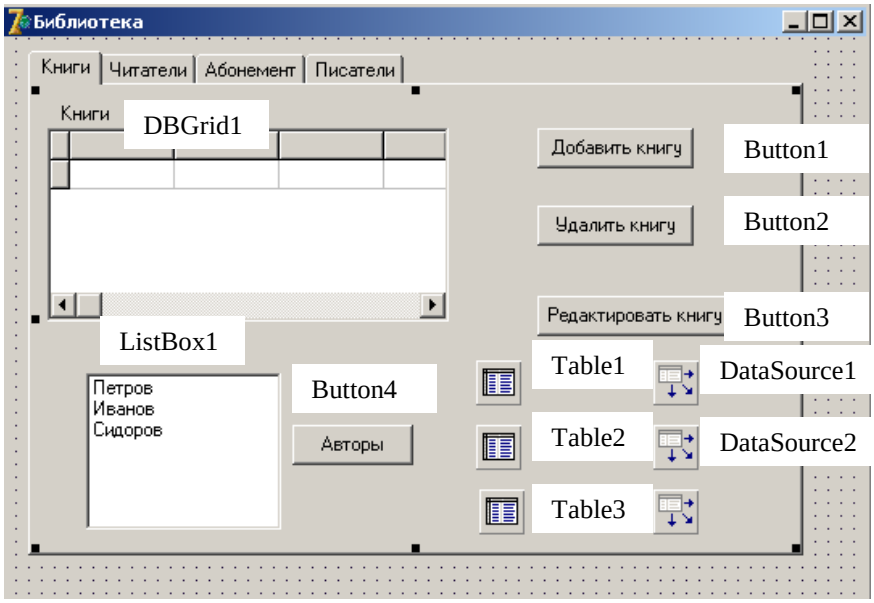
Разрабатываемое приложение должно поддерживать операции добавления, удаления и редактирования книг, авторов и читателей, обслуживать выдачу и возврат книг читателями.

Разработка приложения начинается с определения пользовательского графического интерфейса. В нашем случае, ввиду большого числа выполняемых приложением функций, в основу графического интерфейса можно положить компонент **PageControl** (набор страниц), являющийся стопкой наложенных друг на друга страниц с корешками. Корешки позволяют выбирать текущую (видимую) страницу.

Поместите на форму объект **PageControl1** и создайте четыре страницы: **Книги, Читатели, Абонемент, Писатели**. Первоначально объект **PageControl1** пуст. Для добавления новой страницы надо выполнить команду **New Page** контекстного меню. Появляется новый объект **TabSheet1** (страница), в клиентской области которого можно размещать любые компоненты **Delphi.** Название страницы определяется свойством **Caption** объекта **TabSheet1**.

Страница «Книги».

Разместите на странице указанные на рисунке компоненты. С помощью инспектора объектов установите связь компонентов **Table1, DataSource1** и **DBGrid1** с таблицей **Books**. Свяжите **Table2** и **DataSource2** с таблицей **Writers**, а **Table3** с таблицей **Authors**. Запретите редактирование компонента **DBGrid1**, разрешите редактирование наборов данных **Table1, Table2, Table3.**



Компонент **DBGrid1** позволяет просматривать имеющиеся в библиотеке книги, авторы книг показываются в списке **ListBox1** при нажатии кнопки **Button4.**

Для определения авторов выбранной в **DBGrid1** книги используется связи таблиц **Books, Writers** и **Authors**:

Books ← Authors ← Writers

{(главная) (подчиненная)}

{(главная) (подчиненная)}

Таблица **Books** является главной для **Authors**, а **Authors** – главной для **Writers**. Связи устанавливаются по полям **BkId** (**Books←Authors**) и **WrId** (**Autors←Writers**).

Первая связь позволяет выделить в отношении **Authors** все идентификаторы писателей, являющихся авторами книги, а вторая помогает определить их имена. Сделайте наборы данных активными.

Напишите обработчик события **OnClick** кнопки **Button**4.

```
procedure TForm1Button1Click (…);
Begin
  ListBox1.Clear;
  While not (Table3.EOF) do
    Begin
```

Поместить текущего автора

(Table2.FieldByName('Name').AsString)

В список ListBox1 (ListBox1.Items.Add(…))

Table3.Next;

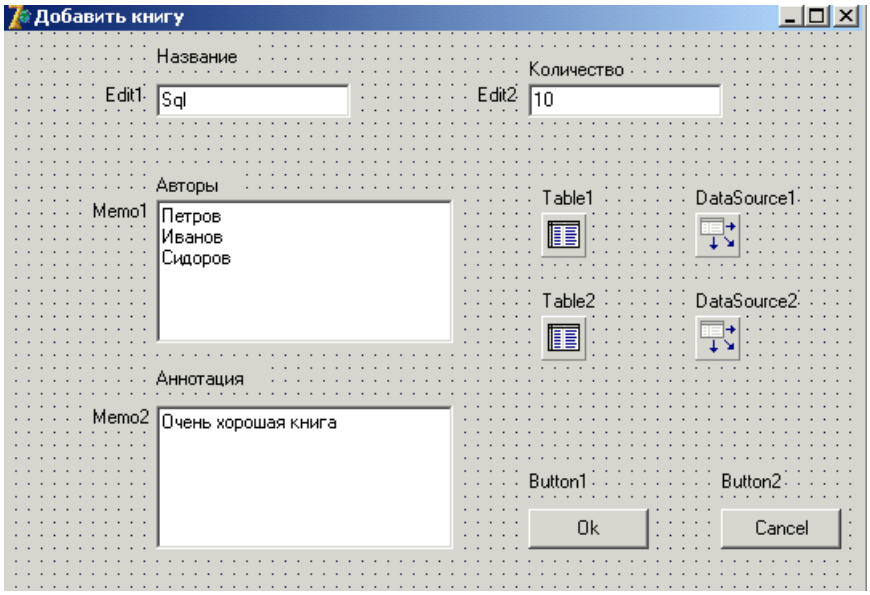
End;

End;

Сохраните проект и проверьте его работу. Придумайте способ очистки **ListBox1** при изменении положения указателя выбранной книги в **DBGrid1** до нажатия кнопки **Button4**.

Задание 3. Добавление новой книги.

Добавьте в проект новую форму **Form2** и установите ее свойства **Caption**= «Добавить книгу», **BorderStyle**= bsDialog. Форма **Form2** будет использоваться как модальное окно, запускаемое на выполнение методом **ShowModal**. Разместите на форме управляющие элементы. Название книги вводится в компонент **Edit1**, имена авторов в компонент Memo1. Каждый очередной автор начинается с новой строки (для перехода на новую строку использовать Enter). Аннотация заносится в объект **Memo2**. Информация сохраняется в БД нажатием кнопки **Button1**. Свяжите **Table1** с таблицей **Writers**, а **Table2** с **Authors**. Сделайте **Table1** главной, а **Table2** – подчиненной. Откройте наборы данных **Table1, Table2**.



Обработчик

Button1.

Procedure TForm2.Button1Click (…);

Begin

Проверить наличие книги в т. Books (Form1.Table1.Locate('Name',Edit1.Text,[]))

Если книга есть, то выход (Exit);

Form1.Table1.Append;

Form1.Table1.FieldByName('Name').AsString:=Edit1.Text;

…….. сформировать все поля записи, кроме BkId ……

Form1.Table1.Post;

For I:=0 to Memo2.Lines.Count-1 do

begin

проверить наличие очередного автора в т. Writers

если автора нет, то добавить новую запись в т.Writers

добавить новую запись в т. Authors

подтвердить добавления

end;

Информация для полей новых записей берется из элементов форм и из текущих уже сформированных записей.

Сохраните проект. Проверьте его работу и устраните ошибки.

Задание 4. Удаление книги

Добавьте в проект обработчик для кнопки «Удалить книгу».

Procedure TForm1.Button2Click(…);

Begin

Сформировать строковую переменную str (имена всех авторов книги)

If MessageDlg('Вы действительно хотите удалить книгу' + Table1. FieldByName ('Name'). AsString + 'авторов' + str, mtConfirmation' [mbYes, mbNo] , 0) = mrYes then

Begin

For I:= 0 to Table3.RecordCount-1 do Table3.Delete;

Table1.Delete;

End;

Сохраните проект, проверьте его работу.

Задание 5. Редактирование книги

Добавьте в проект новую форму **Form3,** аналогичную форме **Form2**, и настройте её как **Form2**. Напишите обработчики **CreateForm** и **Button1Click** для **Form3**.

Procedure TForm3.CreateForm (…);

Begin

Edit1.Text:=Table1.FieldByName('Name').AsString;

Memo1.Lines:=Form1.ListBox1.Items.Text;

Memo2.Lines:= Form1.Table1.FieldByName('Note').AsString;

End;

Procedure TForm3.Button1Click (…);

Begin

Изменить название, аннотацию

Удалить связи книги с авторами

Для каждого автора

Если автора нет в т. Writers, добавить автора в т. Writers

Добавить связь автора с книгой

End;

Сохраните и выполните проект.