Лабораторная работа № 1

Тема: Проектирование приложений для работы с базой данных формата Paradox

Цель работы: Получить навыки построения таблиц формата Paradox и их подключения к проекту Delphi.

Описание хода выполнения работы

Задание 1. Создание базы данных «Телефонная книжка»

1. Запустите программу DataBase Desktop 7.0. Задайте команду меню File|New|Table. В результате появится окно Create Table со значением в списке форматов таблицы Paradox 7.0. Так как создавать таблицы будем в формате Paradox 7, нажмите кнопку ОК в этом окне. В результате появится окно Create Paradox 7 Table (Untitled), в котором будет определяться структура таблицы. Вначале создайте таблицу People.

2. Задайте поля таблицы, согласно структуре:

Ключ	*Имя поля*	*Тип*	*Размер, байт*	*Дополнительно*	*Описание*
#	IDPeople	Счетчик		Обязательное, ключевое	Идентификатор записи
	Family	Строка	30	Обязательное	Фамилия
	Name	Строка	15		Имя
	SecName	Строка	15		Отчество
	Birthday	Дата			День рождения
	S	Логическое		Обязательное	Пол
	Notes	Memo	100		Комментарий

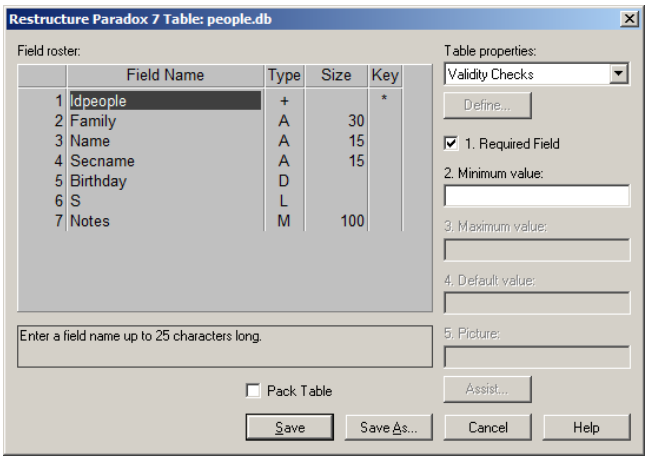
2.1. Введите в первой записи описания структуры, в столбце Field Name, значение IDPeople и нажмите клавишу *Tab* для перехода к следующему столбцу.

2.2. Далее в столбце Type нажмите клавишу пробела и выберите из списка значение +(Autoincrement), затем нажмите клавишу *Tab* для перехода к следующему столбцу.

2.3. В следующем столбце Key нажмите клавишу пробела для указания, что данное поле ключевое (в столбце Key появится звездочка).

2.4. Включите с помощью мыши выключатель Required field (Обязательное поле), а затем выберите мышью поле Key в последней записи и нажмите клавишу *Tab* для перехода к следующей строке, описывающей новое поле таблицы БД.

Остальные поля таблицы задайте аналогичным образом, выбирая при этом соответствующий тип; только для полей Family, Name, SecName и Notes в столбце Size укажите соответствующий размер, а выключатель Required field включите только для полей Family и S. Следует отметить, что все оставшиеся поля не будут ключевыми, а следовательно и поле Key для них не будет активным. Для поля Birthday установите параметр Picture (Шаблон) следующим образом: ## / ## / ####.



3. Сохраните таблицу под именем People.db на диске при помощи кнопки Save As

4. Создайте теперь таблицу Tel, согласно структуре:

Ключ	*Имя поля*	*Тип*	*Размер, байт*	*Дополнительно*	*Описание*
#	IDTel	Счетчик		Ключевое	Идентификатор записи
	IDPeople	Число	4	Обязательное индексированное	Идентификатор человека
	Number	Строка	9	Обязательное	Номер телефона
	TypeTel	Строка	4		Тип телефона

Для этого выполните действия аналогичные действиям, которые выполнялись при создании таблицы People, вводя соответствующие таблице Tel поля, за одним лишь исключением - здесь необходимо создать вторичный индекс по полю IDPeople, который создается следующим образом:

- Выберите в выпадающем списке Table Properties свойство Secondary Indexes.
- Нажмите появившуюся кнопку Define.
- В окне Define Secondary Index в списке Field выберите значение IDPeople, затем нажмите кнопку со стрелкой вправо рядом со списком, после чего нажмите кнопку OK.
- В появившемся окне Save Index As введите в строку ввода *IDPeopleIndex* и нажмите кнопку OK.

5. Сохраните таблицу под именем Tel.db.

Задание 2. Создание проекта для работы с базой данных «Телефонная книжка»

1. Откроем новый проект с помощью команды главного меню File|New|Application. Изменим свойство *Caption* (Заголовок) формы Form1 на «Информация о людях». Фактически этот заголовок формы будет являться названием БД.

2. Сразу же сохраним форму Form1 в файле MyExunitDB.pas, а сам проект –в файле MyExampleDB.dpr. Для этого выполним команду главного меню File|Save |Project As. При этом также появится окно выбора имени файла, в котором выберем для модуля папку, зададим модулю имя MyExUnitDB, после этого зададим проекту имя MyExampleDB.

3. Поместим на форму Form1 компоненты для связи с таблицей БД, находящейся в файле People.db.

3.1. В палитре компонент откроем страницу BDE и перенесем с помощью мыши на форму Form1 компоненту Table (Таблица). Затем выберем свойство *TableName* и для него зададим значение People.db, устанавливая связь приложения с таблицей БД People.db. Для того чтобы отображаемые компоненты могли выводить на экран данные из этой таблицы, зададим свойству *Active* компоненты Table1 значение True.

3.2. Со страницы DataAccess палитры компонент с помощью мыши перенесем на форму Form1 компоненту DataSource (Источник данных). Она обеспечивает связь отображаемых компонент с набором данных. Зададим для свойства *DataSet* компоненты DataSource1 значение Table1 . После этого отображаемые компоненты, размещение которых в проекте будет описано ниже, получат доступ к данным из таблицы БД People.db. Расположение компонент в модуле не существенно.

4. Поместим на форму Form1 компоненты для связи с таблицей БД, находящейся в файле Tel.db, для чего выполним операции, аналогичные операциям в предыдущем пункте, только компоненты Table2 и DataSource2 будут связаны с таблицей Tel.db через свойство *TableName* и *DataSet* соответственно.

5. Установим взаимосвязь между наборами данных для создания формы с подформой, отображающей данные из двух таблиц БД.

Для этого выделим компоненту Table2 на форме Form1 . Установим для ее свойства *MasterSource* значение DataSource1 . Активизируем мышью значение свойства *MasterFields* этой же компоненты. В комбинированной строке ввода Available Index (Используемые индексы) выберем значение IDPeopleIndex, т.к. взаимосвязь таблиц устанавливается с помощью индексированных полей. Затем в левом списке выберем значение IDPeople (поле в подчиненной таблице Tel, в данном примере оно единственное) и в правом списке выберем значение IDPeople (поле в главной таблице People). Нажмем кнопку Add - установленное соединение отображается в нижней части окна. Если установлена ошибочная связь, ее следует выделить в нижнем окне и нажать кнопку Delete, находящуюся справа от окна.

Завершается настройка взаимосвязи таблиц нажатием кнопки ОК. В результате свойство *IndexName* включает название вторичного индекса - IDPeopleIndex, а *MasterFields* содержит имя поля в главной таблице, участвующего во взаимосвязи, - IDPeople.

6. Поместим в форму Form1 компоненты, предназначенные для отображения данных из БД. Для этого выполним следующие действия:

6.1. Со страницы Standard палитры компонент поместим в левый верхний угол формы Form1 метку Label (Метка). Изменим с помощью инспектора объектов ее свойство *Caption* на «Фамилия, имя, отчество».

6.2. Используя страницу Data Controls палитры компонент, поместим в форму Form1 3 компоненты DBEdit (Строка ввода значений в БД) под меткой Label1 в один ряд. Для всех компонент зададим свойству *DataSource* значение DataSource1, а в свойстве *DataField* для каждой из этих строк ввода выберем соответственно из списка значения Family, Name и SecName.

6.3. Со страницы Standard палитры компонент поместим в форму еще одну метку Label. Разместим ее под компонентой DBEdit1. Изменим ее свойство *Caption* на «День рождения».

6.4. Правее этой метки со страницы Data Controls палитры компонент поместим строку ввода DBEdit. Зададим ее свойству *DataSource* значение DataSource1 , а в свойстве *DataField* выберем из списка значение Birthday.

6.5. Правее этой строки ввода со страницы Data Controls палитры компонент поместим компоненту DBCheckBox (Выключатель значения в БД). Изменим свойство *Alignment* (Расположение) у компоненты DBCheckBox1 На LeftJustify, а *Caption* - на «Пол». Затем зададим ее свойству *DataSource* значение DataSource1 , а в свойстве *DataField* выберем из списка значение S.

6.6. Ниже метки Label2 поместим со страницы Standard палитры компонент еще одну метку Label. Изменим ее свойство *Caption* на «Комментарий».

6.7. Под этой меткой поместим со страницы Data Controls палитры компонент компоненту DBMemo (Редактор в БД). Увеличим ее по горизонтали на всю величину формы, а вертикальный размер несколько уменьшим. Зададим ее свойству *DataSource* значение DataSource1, а в свойстве *DataField* выберем из списка значение Notes.

6.8. В правую верхнюю часть формы Form1 поместим со станицы DataControls компоненту DBNavigator (Навигатор по записям БД). Записью в таблице БД является строка. Зададим свойству *DataSource* значение DataSource1.

7. Поместим в форму Form1 со страницы Data Controls компоненту DBGrid (Таблица отображения БД). Поместим ее ниже всех компонент, уже размещенных в форме, и несколько уменьшим по вертикали. Здадим ее свойству *DataSource* значение DataSource2. Если все указанные выше действия были выполнены верно, в компоненте DBGrid1 сразу будет выведено содержимое таблицы Tel.db.

8. Определим список полей, значения из которых следует отобразить в компоненте DBGrid. Для этого необходимо выполнить следующие действия:

8.1. Активизируем мышью свойство *Columns,* тем самым вызвав редактор списка полей таблицы Editing DBGrid 1.Columns.

8.2. В этом редакторе нажмем кнопку Add. Выделим появившееся название «0-TColumn». С помощью инспектора объектов дадим свойству *FieldName* значение Number. Для сложного свойства *Title* дадим его параметру *Caption* значение Номер.

8.3. В окне редактора списка поле нажмем кнопку Add. Выделим появившееся название «1 - TColumn». С помощью инспектора объектов дадим свойству *FieldName* значение TypeTel. Активизируем свойство *PickList* и в появившемся текстовом редакторе введем две строки:

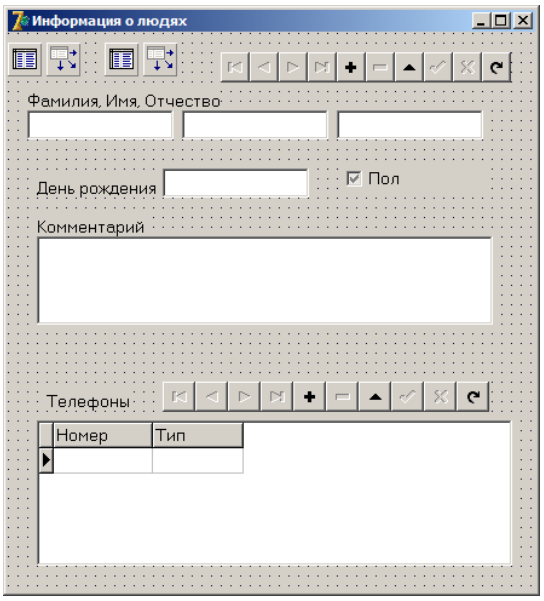
дом.
раб.

После этих операций нажмем кнопку ОК. Таким образом мы задали список, из значений которого можно определять тип телефона. Затем для сложного свойства *Titel* дадим его параметру *Caption* значение Тип. Можно заметить, что отображаемые столбцы появляются в компоненте DBGrid1 по мере их определения.

8.4. Чтобы завершить работу с редактором столбцов таблицы, достаточно его закрыть, воспользовавшись стандартной кнопкой на границе окна.

9. Со страницы Standard палитры компонент поместим в форму Form1 метку Label. Разместим ее над компонентой DBGrid 1 в левой части. Изменим свойство *Caption* на Телефоны.

10. Перенесем со страницы DataControls в форму Form1 еще один навигатор DBNavigator. Разместим его над компонентой DBGrid1 в правой части. Зададим свойству *DataSource* этого навигатора значение DataSource2.



11. Запустим программу с помощью главного меню Run|Run.

12. Нажмем кнопку «Плюс» на панели верхнего навигатора по записям БД.

После этого создается новая запись в таблице People, куда внесем информацию о новом человеке.

В три строки ввода под меткой «Фамилия, имя, отчество» последовательно введем значения «Хромов», «Олег», «Львович». В строку ввода рядом с меткой «День рождения» введем 14.01.79. Выключатель с надписью «Пол» должен быть включен. Нажмем кнопку «Галочка» на верхней панели навигатора по записям. Созданная запись о человеке сохраняется в таблице People.

Сохранить данные о человеке необходимо до начала ввода его телефонов, иначе они не будут соответствовать этому человеку, а в БД окажется столько лишних записей, сколько в этом случае введено телефонов.

12.2. Нажмем кнопку «Плюс» на панели нижнего навигатора по записям. Создается новая запись в таблице Tel. Введем в ячейку «Номер» номер телефона. Выберем в ячейке «Тип», элемент «дом.» из выпадающего списка.

12.3. Нажмем кнопку «Плюс» на панели нижнего навигатора по записям. Кнопку «Галочка» нажимать для сохранения записи необязательно: при вводе последовательно нескольких телефонов и переходе на новую запись данные о телефоне автоматически сохраняются в таблице, и в этой же таблице создается новая запись. Введем информацию еще об одном телефоне. Нажмем кнопку «Галочка» на панели нижнего навигатора по записям. Созданная запись о телефоне сохраняется в таблице Tel.

13. Завершим работу программы нажатием стандартной кнопки закрытия окна на границе формы.

Задание 3. Работа с редактором полей при создании формы БД «Телефонная книжка»

1. Будем выводить дополнительно порядковые номера записей (значение поля IDPeople). При этом будем выводить в виде трехзначного числа, независимо от фактического количества цифр, и набжать выведенную информацию пояснением «Порядковый номер». Для этого выполним следующие действия.

1.1. На форме Form1 активизируем компоненту Table1. В появившийся редактор полей перенесем все поля таблицы, используя команду локального меню редактора Add Fields. В списке полей, появившемся в редакторе, выделим поле IDPeople. В инспекторе объектов появятся характеристики этого поля. В свойство *DisplayFormat* поместим следующую строку: **000**

1.2. Используя страницу DataControls палитры компонент, поместим в верхний левый угол формы Form1 , перед навигатором, компоненту DBEdit. Несколько увеличим компоненту по горизонтали, чтобы в ней полностью разместилась выводимая информация. Используя инспектор объектов, зададим ее свойству *DataSource* значение DataSource1 , а в свойстве *DataField* выберем из

списка значение IDPeople. Если операции выполнены правильно, в строке ввода DBEdit5 должна появиться соответствующая информация.

2. В предыдущих примерах мы выводили поле «Пол» с помощью компоненты DBCheckBox1 , что весьма не наглядно. Выведем это значение словами «Мужской» или «Женский». Для этого выполним следующие операции:

2.1. На форме Form1 активизируем компоненту Table1. В списке полей выделим поле S. В инспекторе объектов появятся характеристики этого поля. В свойство *DisplayValues* поместим следующую строку: Мужской;Женский

2.2. Удалим из формы Form1 компоненту DBCheckBox1 , выделив ее и нажав клавишу *Del*. Используя страницу DataControls палитры компонент, поместим на ее место компоненту DBComboBox (Комбинированная строка ввода значений в БД) и слегка уменьшим ее размер по горизонтали. Используя инспектор объектов, зададим ее свойству *DataSource* значение DataSource1 , а в свойстве *DataField* выберем из списка значение S. Если операции выполнены правильно, в строке ввода DBComboBox1 должна появиться соответствующая информация («Мужской» или «Женский»). В инспекторе объектов для компоненты DBComboBox1 активизируем мышью свойство *Items* и в появившемся редакторе строк введем следующие строки:

Мужской
Женский

Работу с редактором завершим нажатием кнопки ОК.

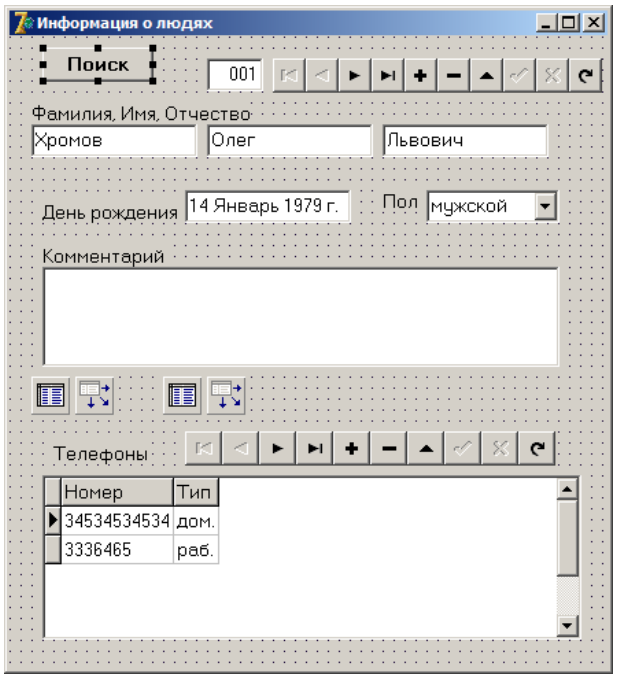
2.3. Используя страницу Standard, поместим слева от комбинированной строки ввода компоненту Label. Используя инспектор объектов, зададим ее свойству *Caption* значение «Пол ».

3. Теперь выберем формат даты для вывода дня рождения человека. Для этого выполним следующие действия:

3.1. На форме Form1 активизируем компоненту Table1. В списке полей выделим поле Birthday. В инспекторе объектов появятся характеристики этого поля.

3.2. Выделим в инспекторе объектов свойство *DisplayFormat* этого поля и введем в него какой-либо формат, например «dddddd». Завершим ввод формата нажатием клавиши *Enter*.

4. Запустим программу с помощью команды главного меню Run|Run. На экране будут отображены все внесенные нами изменения.



5. Перемещаясь по записям, отметим, что в добавленной строке ввода появляется порядковый номер записи, соответствующим образом отформатированный. А в комбинированной строке ввода помещается значение пола человека, выраженное соответствующим словом. Это значение при необходимости можно скорректировать, используя выпадающий список.

6. Занесите в базу данных информацию о 5 – 10 людях.

Задание 4. Организовать процедуру поиска информации в базе.

1.Добавьте на форму компонент *button1*, в свойстве *caption* которого напишите *Поиск по фамилии*.

2. В обработчик события OnClick для кнопки поиск напишите процедуру:

```
procedure TForm1.Button1Click(Sender: TObject);
 var s:string;
begin
InputQuery('Поиск по фамилии','введите фамилию',s);

{ищем запись в таблице, если такая есть, то переводим курсор на нее. Параметр
loCaseInsensitive позволяет вести  поиск  без учета регистра}

table1.Locate('Family',s,[loCaseInsensitive]);

{если информация не найдена, выводим соответствующее сообщение}

if table1.Locate('Family',s,[loCaseInsensitive])=false
     then ShowMessage('Информация не найдена!');
end;
```

3. Сохраните проект. Запустите его и проверьте работу кнопки, вводя различные значения.

4. Самостоятельно поместите на форме кнопки, позволяющие осуществлять поиск информации по д*ате рождения*, *фамилии и имени*.

Примечание.

Для поиска по нескольким полям используйте метод Locate в виде:

```
table1.Locate('Family;Name',VarArrayOf([s,d]),[loCaseInsensitive]);
```

где s – искомая фамилия, d – искомое имя.