Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный университет геодезии и картографии»
(МИИГАиК)

М Е Т О Д И Ч Е С К И Е У К А З А Н И Я

по дисциплине «Методы и средства проектирования ИС и технологий»

Расчетно-графическая работа №3

«Разработка интерфейса информационной системы справочного назначения в среде Delphi. Навигационный доступ к данным»

Для студентов IV курса
Направление подготовки: 09.03.02 «Информационные системы и технологии»
Профиль подготовки: «Геоинформационные системы»

МОСКВА 2016

УДК 681.3(075)

Составитель: Лапчинская М.П.

Методические указания по выполнению расчетно-графической работы № 3 «Разработка интерфейса информационной системы справочного назначения в среде Delphi. Навигационный доступ к данным». Изд. 2, обновл. – М.: Изд. МИИГАиК, 2016. – 20 с.

Методические указания написаны в соответствии с утвержденной программой дисциплины «Методы и средства проектирования информационных систем и технологий» в разделе проектирования и разработки программной части информационных систем для студентов IV курса дневного отделения факультета прикладной космонавтики и фотограмметрии, рекомендованы кафедрой информационно-измерительных систем и утверждены к изданию редакционно-издательской комиссией факультета прикладной космонавтики.

Методические указания по выполнению данной расчетно-графической работы призваны помочь студентам при самостоятельной разработке практической части курсовой работы одноименного курса.

Методическое пособие может быть использовано студентами, магистрами и аспирантами других факультетов и специальностей.

Рис. 20, библиогр. 6 назв.

Рецензенты:

к.т.н. Ребрий А.В., зам. ген. директора ООО «ЭСТИ МАП».

к.т.н., Кондауров И.Н., доцент кафедры «Информационно-измерительные системы» Московского государственного университета геодезии и картографии (МИИГАиК)

© Московский государственный университет
геодезии и картографии, 2016

Введение

Одной из базовых и заключительных дисциплин по направлению подготовки студентов 09.03.02 Информационные системы и технологии является дисциплина «Методы и средства проектирования информационных систем и технологий».

В настоящих методических указаниях рассматриваются вопросы, изучаемые в дисциплине «Методы и средства проектирования информационных систем и технологий» в разделе проектирования программной части информационных систем.

Целью методических указаний является – обучение студентов на конкретных примерах последовательности типовых действий при разработке интерфейсов информационных систем в среде программирования Delphi, что позволит студентам освоить базовые принципы и методы технологии разработки программной части современных информационных систем и поможет применить их при курсовом проектировании.

Для успешного изучения материала достаточны знания основ программирования на языке программирования Object Pascal и навыков работы в среде программирования Delphi. Желательно ознакомиться с материалом учебных пособий [1-6].

1. Постановка задачи

Разработать интерфейс информационной системы (ИС) справочного назначения «Сотрудники фирмы» в форме клиентского приложения, включающего:

1) структуру реляционной базы данных (БД) в среде MS Access в режиме конструктора таблиц;
2) клиентское приложение – интерфейс на основе драйвера СУБД в среде Delphi: *навигационный* способ доступа к данным;

2. Основные теоретические сведения по реляционным базам данных

2.1. Определение базы данных

Базы данных – структурированное хранилище информации, представленное в виде файла (чаще) или файлов (реже).

В системе управления базой данных (СУБД) **приложение *.exe** (рис.1) выделяют, как правило, в отдельную систему, т.к. она должна уметь выполнять следующие основные функции манипулирования данными:

1. Отображение
2. Добавление
3. Удаление
4. Редактирование
5 Обработка, где главными являются 3 функции:
   а) поиск
   б) сортировка
   в) фильтрация
6. Расчетные функции (как правило, ограниченно используемые):
   а) сумма
   б) среднее
   в) min значение
   г) max значение
   и прочее.

Как осуществляется выполнение данных функций СУБД? – в общем случае посредством механизма связи какого-либо **приложения *.exe** с некоторым **файлом данных** или **базой данных** (**БД**); таким механизмом связи является некоторый **драйвер СУБД** – рис.1:

Общая схема системы управления базой данных (СУБД):

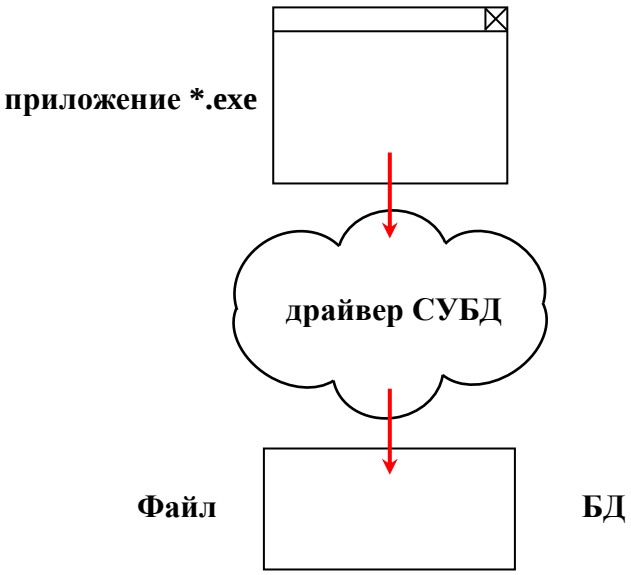


Рис. 1

Что такое драйвер СУБД? – это тоже некоторое exe-приложение, т.е. исполняемый машинный код, но созданный только для функций (классов) на типовые алгоритмы поиска, фильтрации, сортировки и т.п. Таким образом, драйвер СУБД представляет собой также exe-приложение, включающее готовую библиотеку функций, которую подгружает (подключает) приложение *.exe (говорят, динамически подключаемая библиотека).

Для динамически подключаемых библиотек принята аббревиатура **DLL** – **d**inamic **l**ink **l**ibrary – библиотека динамической компоновки.

Обычно к формату хранения данных в БД предъявляют следующие основные требования:

1. Универсальность.
2. Компактность (т.е. хорошо сжимаемый).
3. Оперативность (т.е. скорость выполнения операций, например, добавление информации в файл, должна выполняться быстро).

При выполнении операций с таблицами используется один из следующих способов доступа к данным:

- навигационный способ (в настоящее время используется редко);
- реляционный способ.

2.2. Реляционная база данных (БД)

Существует несколько моделей хранения и манипулирования данными (т.е. способов представления структуры данных, методов защиты данных и операций, выполняемых с данными): иерархическая, сетевая, звездообразная, реляционная модели.

Каждая из этих моделей имеет свои преимущества при решении задач определенного типа. Однако гибкость и эффективность реляционной модели делают ее наиболее популярным инструментом для разработки баз данных.

Таким образом, **реляционная база данных (БД)** основана на реляционной модели представления данных, которая представляет собой средство для рационального и эффективного хранения информации, обеспечивающее:

- надежную защиту данных от случайной потери или порчи (целостность данных);
- экономное использование ресурсов (людских и технических);
- механизмами поиска информации, удовлетворяющими разумным требованиям к производительности.

В основе реляционных баз данных лежат следующие принципы:

- все данные на концептуальном (не на физическом) уровне представляются в виде упорядоченной организации, определенной в виде строк и столбцов и называемой ***отношением*** (*relation*). Термин «отношение» однозначен в отличие от термина «таблица», который имеет множество дополнительных значений; поэтому его и положили в основу наименования базы данных;

- все значения являются ***скалярами***: для любой строки и столбца любого отношения существует одно и только одно отношение;

- все операции выполняются над целым отношением, и результатом выполнения этих отношений также является целое отношение. Этот принцип называется ***замыканием***. Он позволяет непосредственно использовать результаты одной операции в качестве исходных данных для выполнения другой. В частности, драйвер MS Jet, и SQL Server дают возможность использовать результаты одного запроса для составления нового.

Таким образом **реляционная база данных основана на таблицах**, таблица – это соотношение (от англ. relation) как минимум двух видов информации: по горизонтали и по вертикали, что и породило название «Реляционная БД».

Почему информацию помещают в таблицах? – таблица это аналог плоскости, а плоскость позволяет нагляднее всего её представить (сравните хотя бы с каким-либо трехмерным представлением информации, четырехмерным и выше). Следует всегда помнить, что, строго говоря, информация может быть не только плоской (т.е. двумерной –

2D), но и трехмерной – **3D** и выше – **>3D**. Но в любом случае таблицей (т.е. плоскостью) пользоваться удобнее и проще всего. Таким образом, благодаря своему главному достоинству, – простоте, – таблицы и получили среди пользователей наибольшее распространение, примером чего служит среда MS Excel.

Однако реляционные БД имеют и недостаток – слабо поддерживаются математические расчеты. Аналитические расчеты хорошо поддерживаются так называемым OLAP-сервисом [6].

В отличие от таблиц, организуемых в MS Excel, реляционные таблицы должны удовлетворять определенным критериям. Для пояснения этих критериев рассмотрим пример таблицы 1:

Таблица 1

PK **поле**

запись

ID	**Name**	**Salary**
1	Иванов	200$
2	Петров	300$
3	Сидоров	400$
4	Захаров	500$
5	Леонов	600$
6	Петров	700$

1-ый критерий: поле **ID** (**id**entification). В реляционных таблицах отсутствует понятие «текущая запись». Поэтому, чтобы осуществить операцию удаления записи, необходимо указать некоторый признак, например, фамилию Петров (таблица 1). Но записей с фамилией Петров может быть несколько, поэтому, чтобы удалить запись, надо ввести дополнительные признаки, например: имя отчество Петрова, номер его паспорта, ИНН и т.п. Зачастую такие данные отсутствуют, поэтому в записи вводят дополнительное поле **ID** (первые две буквы от англ. **id**entification – идентификация) и в простейшем случае проставляют номера в порядке возрастания (таблица 1). В этом случае структура БД будет работать всегда корректно (нет зависимости от содержания информации), и мы всегда сможем удалить ту запись с фамилией Петров, которую надо.

2-ой критерий: первичный ключ **PK** (**p**rimary **k**ey).

Помимо операции удаления записи существует важная операция добавления записи. При этом существует случайный риск порчи нашей таблицы, например, случайно образуется 2-ой дубль идентификатора **ID**. Поэтому, чтобы избежать дублирования в таблицу вводится некоторый предохранитель, называемый первичным ключом – **p**rimary **k**ey (**PK**), который всегда позволяет добраться до уникальной записи. Именно благодаря **PK** не происходит порчи структуры данных, что осуществляет драйвер СУБД. Таким образом, все современные реляционные таблицы требуют установки **PK** и, более того, если не установить **p**rimary **k**ey (**PK**), то появляется запрос-предупреждение.

2.3. Разработка структуры базы данных в MS Access

Согласно общей схеме СУБД (рис.1) при разработке структуры базы даных в MS Access в качестве:

- приложения *.exe, у нас выступает программная среда MS Access (рис.2);
- файла базы данных – файл *.mdb (рис.2);
- драйвера – MS Jet – специальное приложение, предназначенное для физического манипулирования данными *.mdb,: хранением их на диске и извлечением по SQL-запросу (рис.2).

Программный код Microsoft Access, – некоторая интерактивная среда, с помощью

которой драйверу MS Jet объясняется, что же будут представлять собой данные – таблицы, отчеты, формы и др. В результате создаются физические объекты – файлы с расширением ***.mdb**, в которых эти данные хранятся, как правило, на жестком диске. Таким образом реляционная база данных содержит: физические таблицы, представления, запросы, хранимые процедуры, а также правила, используемые драйвером MS Jet для защиты данных (в частности, primary key (PK) – первичный ключ).

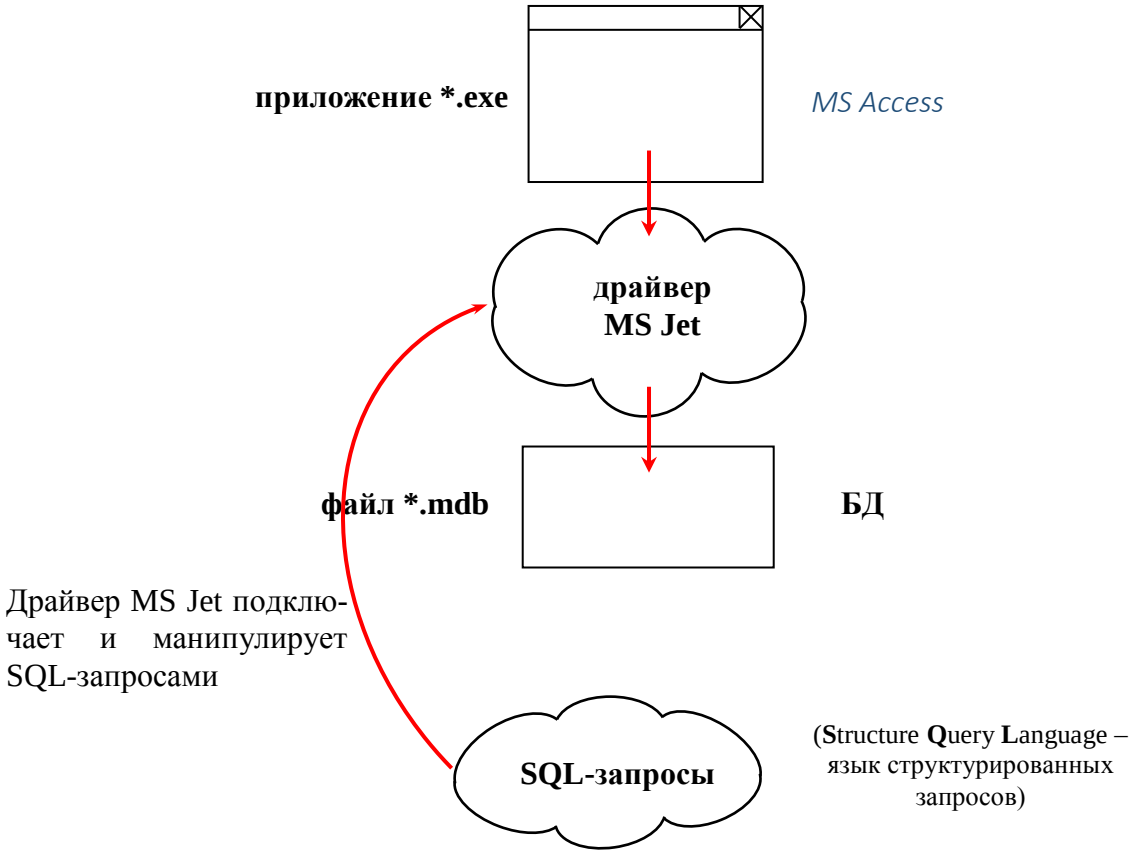


Рис. 2

Замечание: вместо драйвера MS Jet могут быть использованы и другие драйверы, в частности, MS SQL Server; назначение драйверов MS Jet и MS SQL Server одинаково – это инструменты для хранения данных и манипулирования ими, но они существенно различаются по внутренноей архитектуре, а следовательно и по назначению:

MS Jet – это «настольный» сервер баз данных, ориентированный на простейшие информационные системы: малые и средние;

MS SQL Server использует клиент-серверную архитектуру и предназначен для создания информационных систем от средних до больших, т.к. может поддерживать до нескольких тысяч пользователей, работающих с различными приложениями (изучение MS SQL Server 2000 – в на 5 курсе).

В данной расчетно-графической работе интерфейс приложения (раздел 3) будет разрабатываться в два этапа:

I этап: разработка структуры реляционной базы данных (БД) в среде MS Access;

II этап: разработка клиентского приложения – интерфейса на основе драйвера СУБД, содержащего классы типовых алгоритмов в среде Delphi (*навигационный способ доступа к данным*).

3. Разработка интерфейса клиентского приложения (*навигационный способ доступа к данным*)

3.1. Типовые действия при разработке структуры таблицы

1. Войти в программную среду **MS Access**, например: **Пуск → Программы → Microsoft Access**.

2. Возникают варианты работы с базой данных. Отберите параметр **Новая база данных → OK** для создания пустой базы данных, в которую затем могут быть добавлены таблицы, формы и др.

3. Возникает меню **Файл новой базы данных**, где отберите путь сохранения файла, например: **D:\Students\FPKiF4-2Delphi\FPKiF4-2-1\practika3\brigadaN2** и укажите вместо имени по умолчанию (db1) имя **personal_base** – рис.3. (Напоминаем, что расширение для имен баз данных в среде MS Access – ***.mdb**, которое указывать не надо).

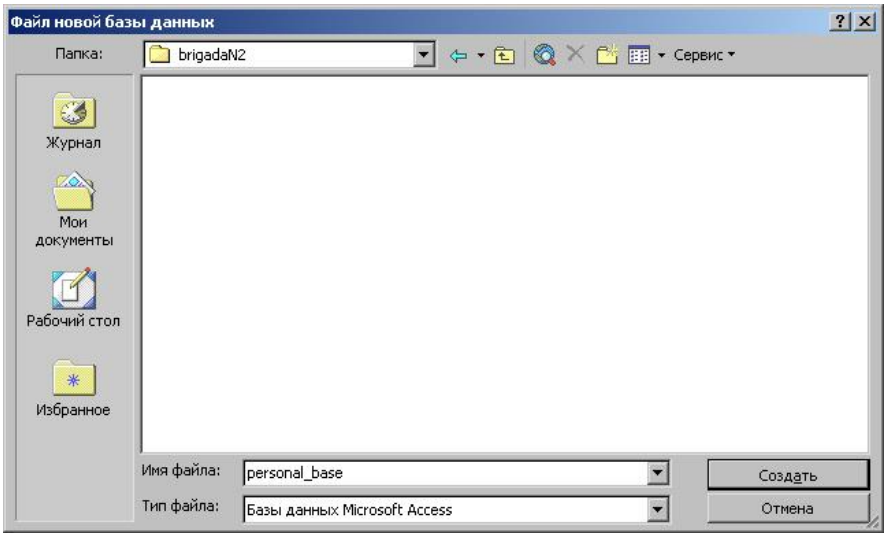


Рис. 3

4. После нажатия кнопки **Создать** возникает меню для работы с файлом **personal_base** – рис.4:

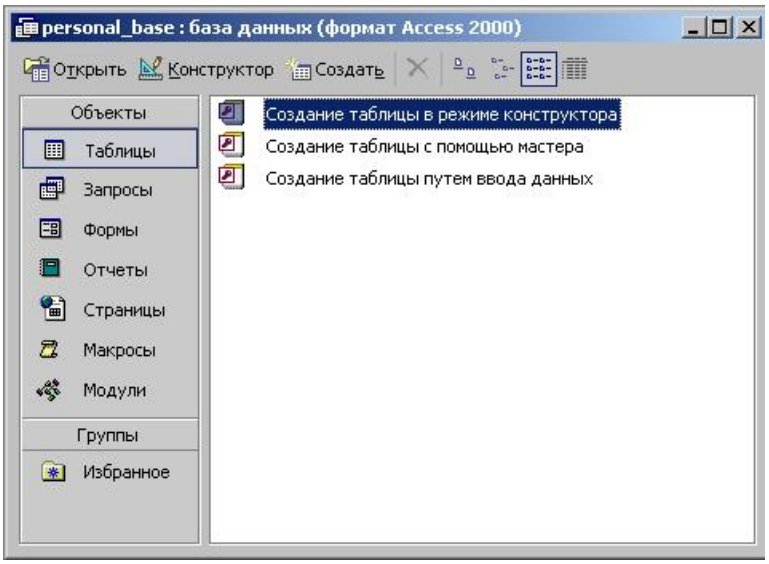


Рис. 4

5. Существует три режима работы с базой данных: 1) в режиме конструктора, 2) с помощью мастера, 3) путем ввода данных. Отберите работу в режиме конструктора: для этого произведите двойной клик по названию режима **Создание таблицы в режиме конструктора** или один клик по пиктограмме **Конструктор**; возникает таблица для создания (конструирования) полей вашей будущей базы данных в виде таблицы – рис.5:

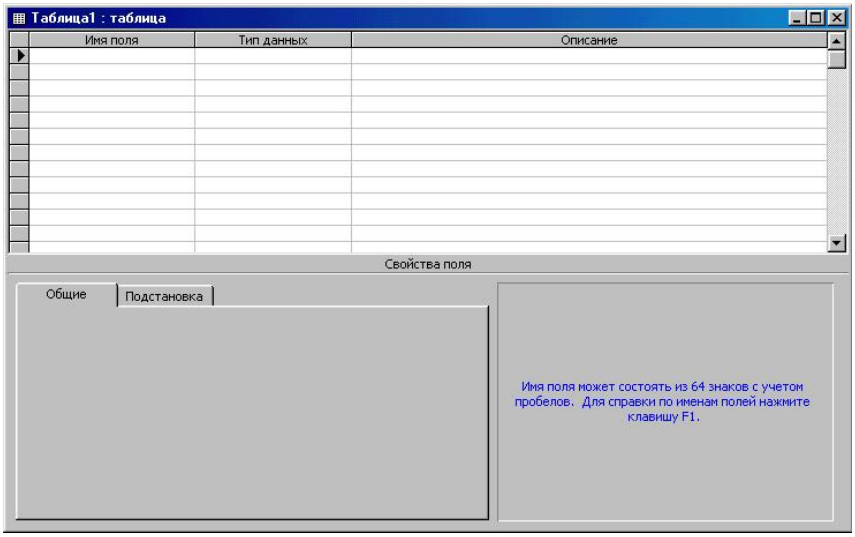


Рис. 5

6. В этой таблице последовательно заполните поля: имя поля и тип данных. Для первого поля (первая строка) введите **personalID** и отберите (по треугольнику выбора) тип поля – **счетчик** – рис.6а:

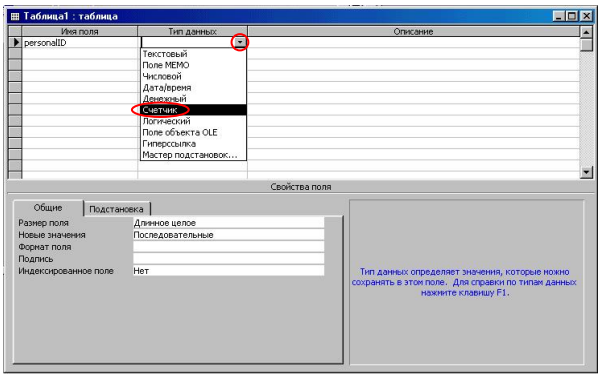


Рис. 6а

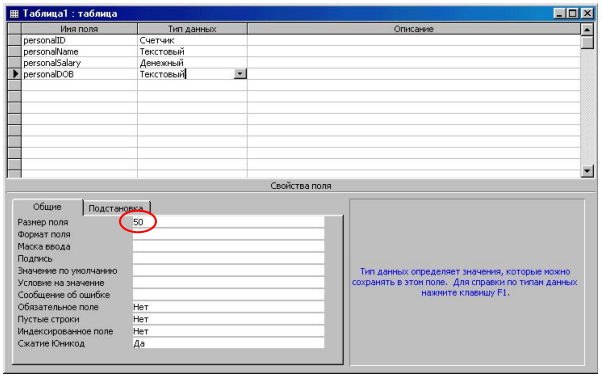


Рис. 6б

7. Аналогично заполните информацией остальные поля и типы данных согласно рис.6б. Названия полей можете ввести и другие, но при этом соблюдайте следующие правила:

1) названия полей набирайте латинскими буквами (не русскими!);
2) если для типа данных **Денежный** вместо денежного значка «**р.**» необходимо установить знак **$**, то войдите **Пуск → Настройка → Панель управления → Язык и стандарты** и во вкладке **Денежная единица** поменяйте в позиции **Обозначение денежной единицы** значок «**р.**» на значок **$**;
3) для типа данных **Текстовый** можете поменять значение длины поля по умолчанию (50),- рис.6б,- на меньшее количество символов (например 20).

В данной расчетно-графической работе рассматривается пример четырех полей, которые описывают следующее содержание (придумайте свое содержание!):

personalID	– Нумерация (идентификатор) персонала какой-либо фирмы
personalName	– Фамилия И.О. (Name) персонала какой-либо фирмы
personalSalary	– Зарплата (Salary) персонала какой-либо фирмы
personalDOB	– Дни рождения (**D**ay **O**f **B**irthday) персонала какой-либо фирмы

Замечания: 1) при наименовании полей употребляйте только латинские буквы, русские буквы не употребляйте, пробелы и знаки дефисов не допустимы;

2) старайтесь для удобства дальнейшей работы с запросами по полям давать мнемонические названия, т.е. названия, напоминающие то или иное содержание поля;

8. Установить курсор на первое поле и нажать в основном меню пиктограмму ключевого поля – первичного ключа **PK** (**P**rimery **K**ey) . – рис.7:

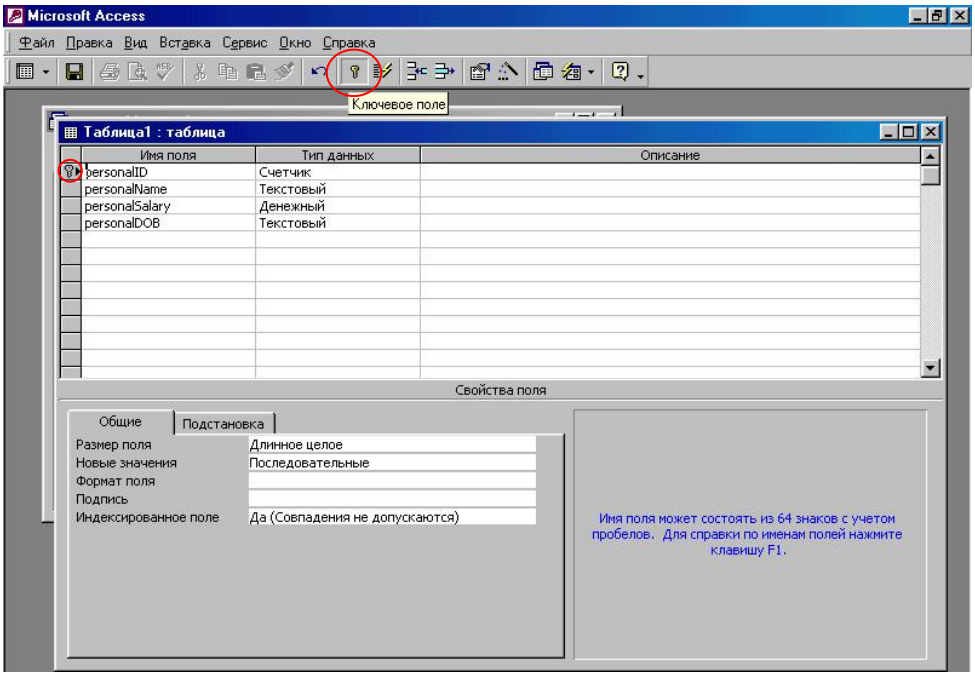


Рис. 7

9. После обязательной установки первичного ключа **PK**, запишите объект таблицы (с именем по умолчанию ‘Таблица1’ – рис. 8а) под именем **personal_tab** – рис. 8б:

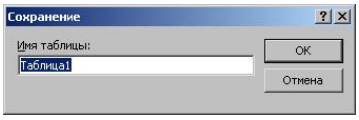


Рис. 8а

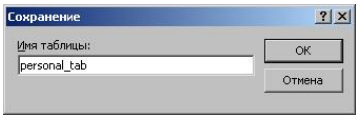


Рис. 8б

10. В результате в базе данных MS Access **personal_base** появится ее экземпляр – таблица с именем **personal_tab** – рис.9:

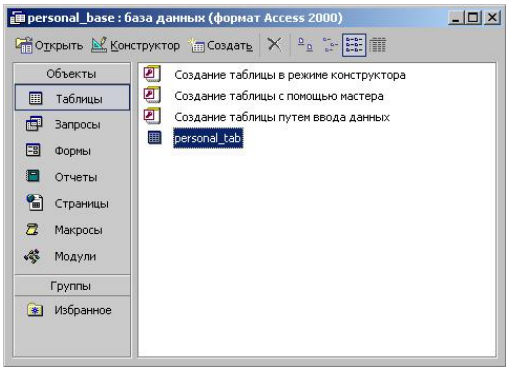


Рис. 9

11. Произведите двойной щелчок по имени таблицы **personal_tab** (рис.9); возникают организованные вами поля таблицы, которые надо заполнить конкретной информацией согласно установленным типам данных – рис.10:

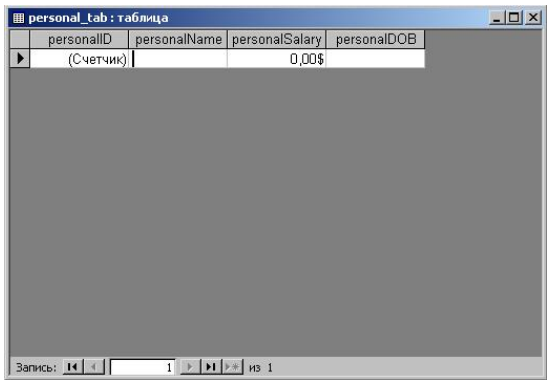


Рис. 10

12. Произведите щелчок сразу по второму полю PersonalName (рис.10) и заполните поля конкретной информацией, например, такой – рис.11. Сохраните таблицу – **Ctrl+S**.

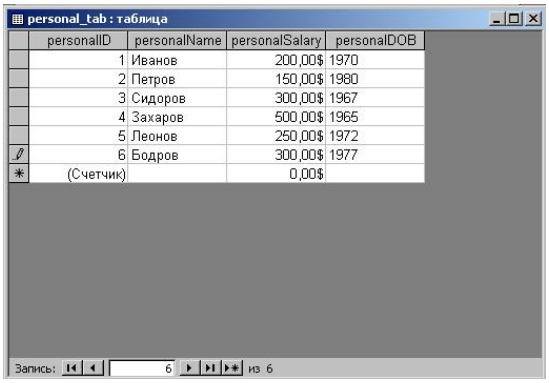


Рис. 11

<u>Замечания</u>: в меню MS Access переход между конструктором таблицы и режимом таблицы можете осуществить двумя основными вариантами:

<u>1 вариант</u>: а) через главное меню – нажмите самую левую пиктограмму;

б) по треугольнику выбора отберите: или режим конструктора
или режим таблицы

<u>2 вариант</u>: а) если вы находитесь в режиме таблицы, то переход в конструктор – по правой кнопке мыши (расположив курсор на синей полосе активного окна) и в контекстном меню отбираем – конструктор таблиц;

б) если вы находитесь в режиме конструктора, то переход в конструктор – по правой кнопке мыши (расположив курсор на синей полосе активного окна) и в контекстном меню отбираем – режим таблицы;

3.2. Разработка клиентского приложения в среде Delphi (*навигационный способ доступа к данным*)

3.2.1. Подготовительные действия

I. Загрузить Delphi с новым проектом: **File→New→Application**

II. Подсветить форму с именем по умолчанию **Form1** и в окне **Object Inspector** (Инспектор объектов) в закладке Properties (Свойства) для параметра **Name** переименовать значение **Form1** в **fmMain**.

III. Нажать (**Save All**) и в открывшемся окне:

1) войти в свою рабочую папку (каталог), например, D:/Students/FPKiF4-2Delphi/FPKiF4-2-1/practika3/brigadaN2;
2) изменить названия по умолчанию модуля и запускающей программы, соответственно: вместо **unit1** ввести **firmpersonals** и далее вместо **project1** ввести **personals**.

IV. Подсветить форму **fmMain** и установить значения для двух свойств:

- для свойства **Caption** заголовок: Информационно-поисковая система “Сотрудники фирмы”;
- для свойства **Icon** загрузить иконку ***.ico** (кнопка **Load…**), которую отберите из папки D:/Students/FPKiF4-2Delphi/Images/Icons, например, **finance.ico**. Если она отсутствует, то отберите иконку ***.ico** из папки D:/Program Files/Common Files/Borland Shared/ Images/Icons. В результате на форме **fmMain** появится иконка **finance.ico** – рис12а. Возможно, также создать собственную иконку *.ico (не более чем 16x16 px) – см. РГР №1.

Нажать быструю кнопку (**Save All**); запустить приложение, например, по быстрой кнопке (**Run**); проверить его работу: должна получиться подготовленная для создания интерфейса чистая форма – рис.12б:

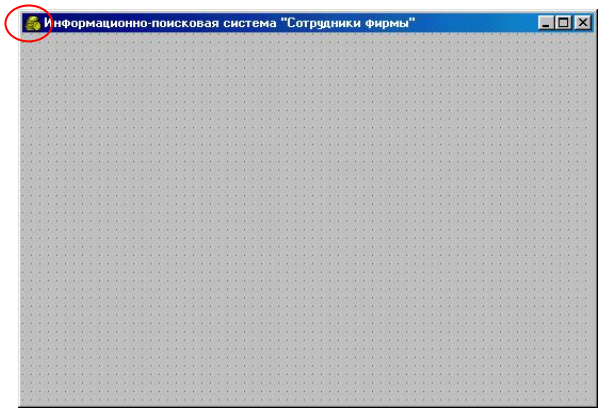


Рис.12а

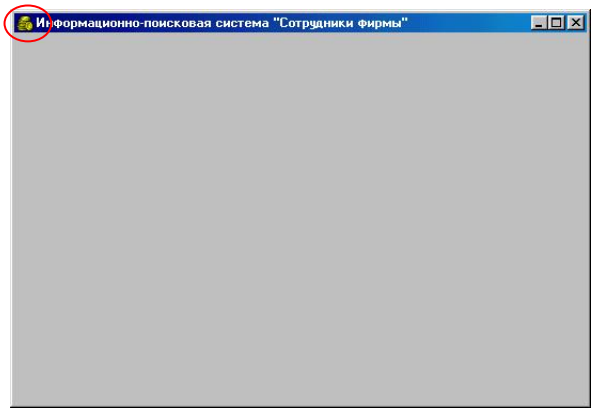


Рис.12б

Обратите внимание, что иконка finance.ico появляется как на форме **fmMain** (рис.12а), так и на приложении **personals.exe** (рис.12б).

3.2.2. Навигационный подход при работе с базой данных

Интерфейс прикладных программ (**A**pplication **P**rogram **I**nterface - **API**) при навигационной работе с базой данных (см. I этап выполнения расчетно-графической работы) создается посредством драйвера **OLE DB** (**O**bject **L**inking and **E**mbedding **D**ate **B**ase – связывание и встраивание объектов базы данных), предоставляющего для среды Delphi различные функции, находящиеся на странице (закладке) **ADO** палитры компонентов (**A**ctiveX **D**ata **O**bjects – объекты данных ActiveX). Компоненты библиотеки ADO предназначены для управления данными и базируются на драйвере OLE DB. В свою очередь посредством драйвера OLE DB осуществляется доступ к любым источникам данных – реляционным таблицам, электронным таблицам, документам в RTF-формате и др.

Таким образом согласно общей схеме СУБД (рис.1) на II этапе создания интерфейса

информационной системы (навигационный подход) в качестве:

- приложения *.exe, у нас выступает программная среда Delphi – интерфейс библиотеки **ADO** (ActiveX Data Objects) – рис.13;
- файла базы данных – файл **personal_base.mdb** – рис.13;
- драйвера – **OLE DB** – специальное приложение, предназначенное для физического манипулирования данными *.mdb, в частности, personal_base.mdb – рис.13.

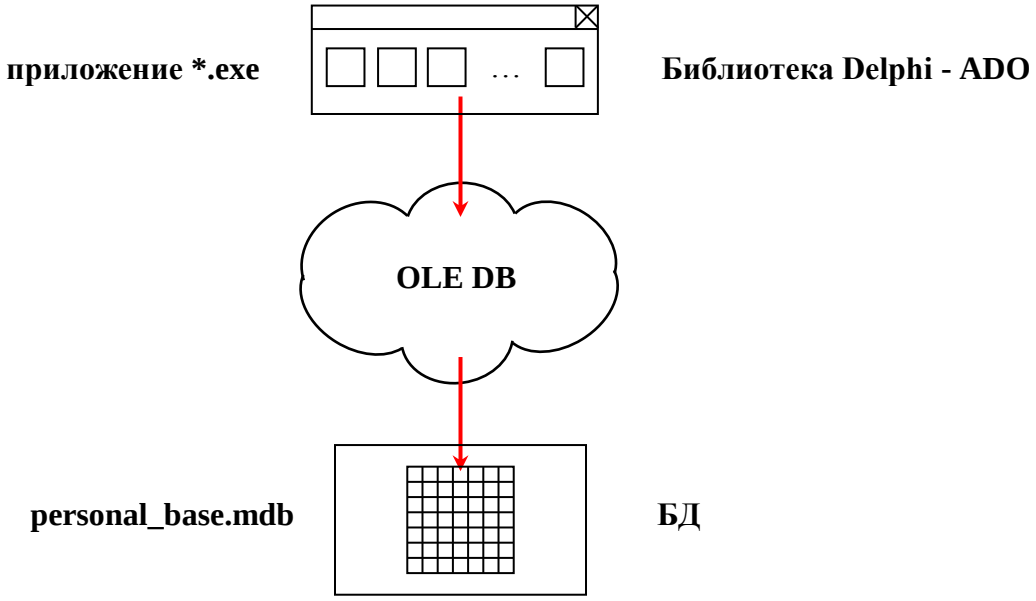


Рис. 13

Программная среда **Delphi**, – интерактивная среда, с помощью библиотеки компонентов которой (закладка **ADO**) драйверу **OLE DB** объясняется, как конкретная таблица (объект) **personal_tab** файла базы данных **personal_base.mdb** будет связана с данной средой, открываться в ней и как в таблице будет организована простейшая навигация.

1. Установить на форму компонент **ADO Table** (закладка **ADO**) – рис.14а:

Рис.14а

В результате на форме появится экземпляр компонента **ADO Table1** вида – рис.14б:

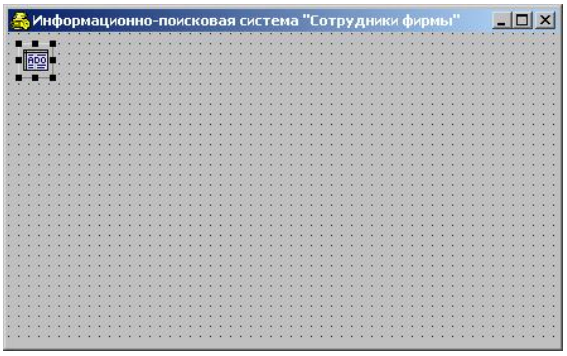


Рис.14б

2. Компонент **ADO Table** необходим для организации соединения с базой данных **personal_base.mdb**. Чтобы этот компонент заработал, необходимо его определенным образом настроить.

Для этого подсветите компонент **ADO Table** и для свойства **ConnectionString** нажмите на троеточие выбора; появляется окно формы **FmMain.ADOTable1 ConnectionString** –

рис.15а:

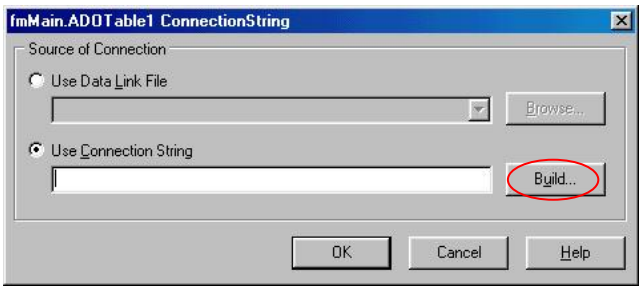


Рис.15а

3. Нажмите **Build** (Встроить) – рис.15а; возникает список драйверов Microsoft, поддерживаемых библиотекой **ADO**, где отберите драйвер **MS Jet 4.0 OLE DB Provider**. – рис.15б:

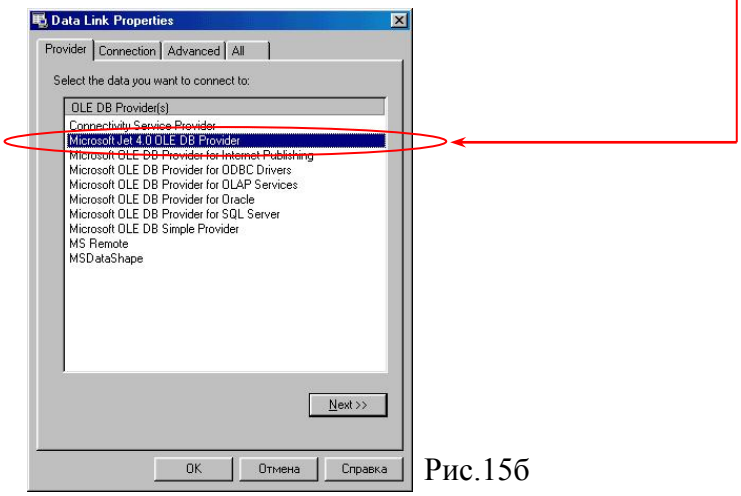


Рис.15б

4. Нажмите **OK** (рис. 15б); возникает текст со строкой используемого соединения – рис.15в:

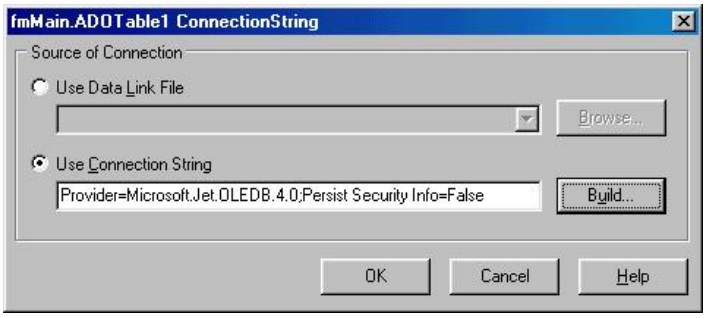


Рис.15в

5. Нажмите кнопку **Build** (рис.15в); возникает вкладка **Connection** (Связь), где в позиции **Select or enter a DataBase Name** по треугольнику выбора установите нашу базу данных **personal_base.mdb** (рис.15д), находящуюся в папке: D:\Students\FPKiF4-2Delphi\FPKiF4-2-1\practika3\brigadaN2 – рис.15г:

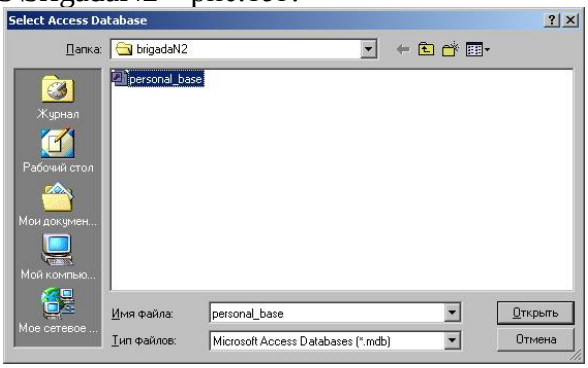


Рис.15г

6. Установите флажок около **Allow saving password** (Разрешить сохранение пароля); в результате вкладка **Connection** (Соединение) будет выглядеть – рис.15д:

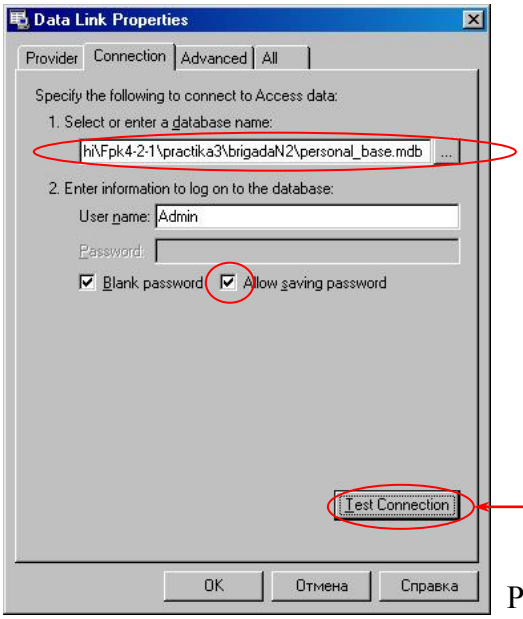


Рис.15д

7. Для проверки успешности связи с файлом базы данных **personal_base.mdb** нажмите кнопку **Test Connection** (Проверка соединения) – рис.15д; в случае успешного подсоединения **personal_base.mdb** возникает надпись: **Test connection succeeded** (Проверка связи успешна) – рис.15е:

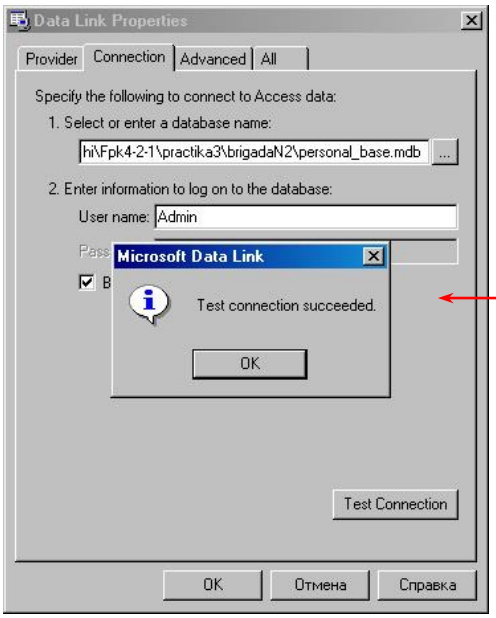


Рис.15е

Нажмите кнопку **OK** – три раза, пока не закроются все открытые окна настройки соединения.

8. Загрузите проводник Windows (если установлен Windows Commander, то используйте его) и в папке, где находится **personal_base.mdb** (D:\Students\FPKiF4-2Delphi\ FPKiF4-2-1\practika3\brigadaN2\personal_base.mdb), через контекстное меню создайте текстовый файл (пустой); откройте его и сохраните с тем же именем, но с расширением **udl**: **personal_base.udl**.

Замечание: при сохранении файла с расширением **udl** (**personal_base.udl**) не забудьте jn отобрать в позиции тип файла: **все файлы (*.*)** и в позиции имя файла указать имя файла **с расширением**.

9. Подсветить на форме **fmMain** экземпляр компонента **ADO Table1** и в **Object Inspector** (Инспекторе объектов) установить значения для 2 свойств:

Свойства (Properties):	**значения:**	
TableName	**personal_tab**	← по троеточию выбора
Active	**true**	← по троеточию выбора

Замечание: соблюдайте последовательность установки значений параметров.

10. Подсветить главную форму **fmMain** и установить на нее экземпляр визуального компонента **DBGrid1** (закладка **DataControls**) – рис.16:

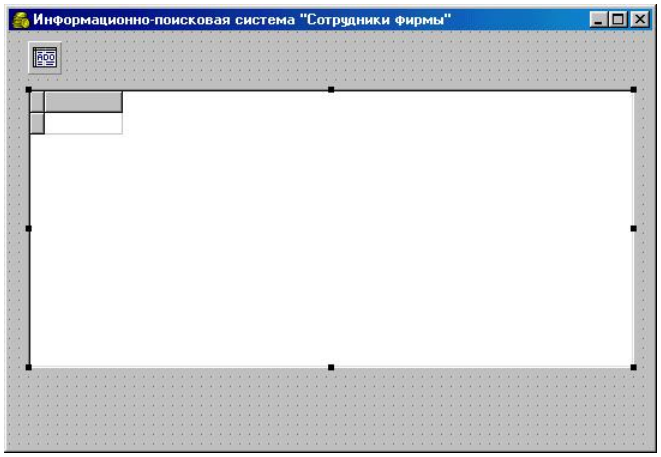


Рис.16

Этот экземпляр компонента необходим для организации структуры таблицы (сетки) и открытия впоследствие в ней нашей конкретной таблицы (объекта) **personal_tab** из файла базы данных **personal_base.mdb** (пока экземпляр таблицы **DBGrid1** – пустой).

11. Подсветить главную форму **fmMain** и установить на нее невизуальный экземпляр компонента **DataSource1** (закладка **DataAccess**) рядом с экземпляром компонента **ADO Table1** (рис.17). Чтобы этот компонент заработал, необходимо его определенным образом настроить.

12. Подсветить на форме **fmMain** экземпляр компонента **DataSource1** и в **Object Inspector** (Инспекторе объектов) установить для свойства (Properties) **DataSet** значение **ADOTable1**. В результате устанавливается связь двух компонентов (говорят - ассоциация): компонента, отвечающего за установку соединения (адресацию связи) с конкретной таблицей (т.е. экземпляром ADOTable1), и компонента, предназначенного для организации доступа к источнику Access-ой базы данных (экземпляр DataSource1).

13. Подсветить на главной форме **fmMain** визуальный компонент **DBGrid1** и в **Object Inspector** (Инспекторе объектов) установить для свойства (Properties) **DataSource** значение **DataSource1**. Тут же в экземпляр пустой таблицы (сетки) **DBGrid1** загружается наша таблица **personal_tab** из файла базы данных **personal_base.mdb** – рис.17:

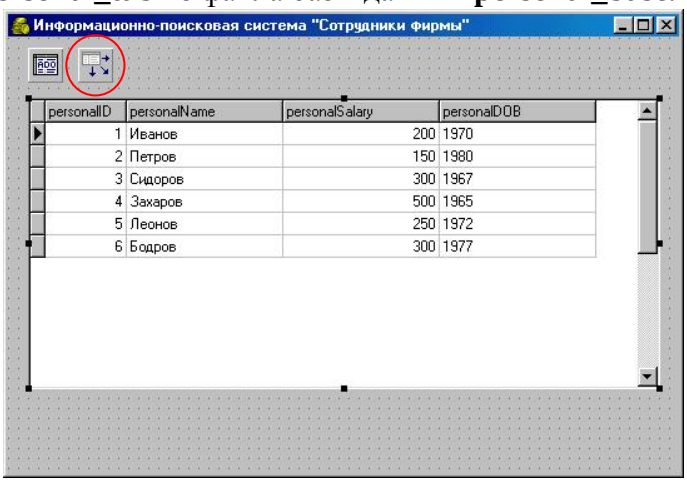


Рис.17

14. Подсветить главную форму **fmMain** и установить на нее визуальный экземпляром компонента навигационного интерфейса **DBNavigator1** (закладка **Data Controls**) – рис.18:

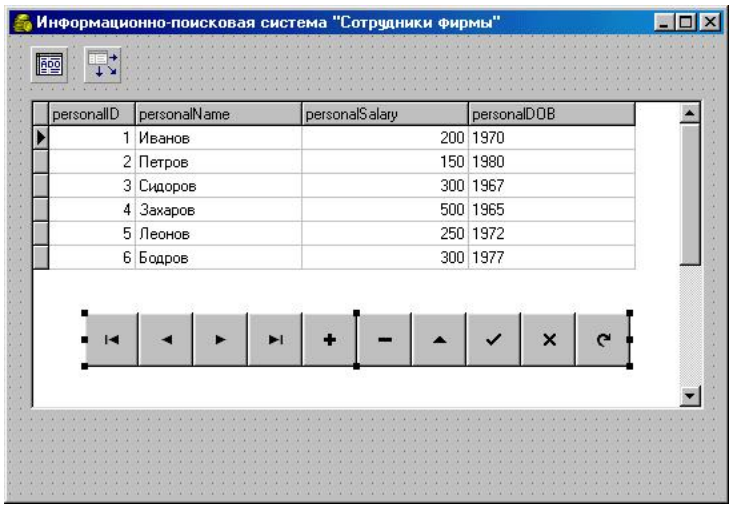


Рис.18

15. Подсветить на главной форме **fmMain** экземпляр компонента **DBNavigator1** и в **Object Inspector** (Инспекторе объектов) установить для свойства (Properties) **DataSource** значение **DataSource1**. Тут же наша таблица **personal_base.mdb** оказывается связанной с экземпляром навигационного интерфейса **DBNavigator1.**

16. Сохраните приложение по быстрой кнопке (**Save All**) и запустите приложение, например, по быстрой кнопке (**Run**) – рис.19.

Проверьте его работу:

- перемещение по навигационному интерфейсу (влево и вправо; в самое начало записей и в самый конец записей) должно вызывать аналогичные перемещения по записям в таблице **personal_tab** из файла базы данных **personal_base.mdb**;

- нажатия на клавиши + и – навигационного интерфейса должны вызывать вставку пустой или удаление существующей строки.

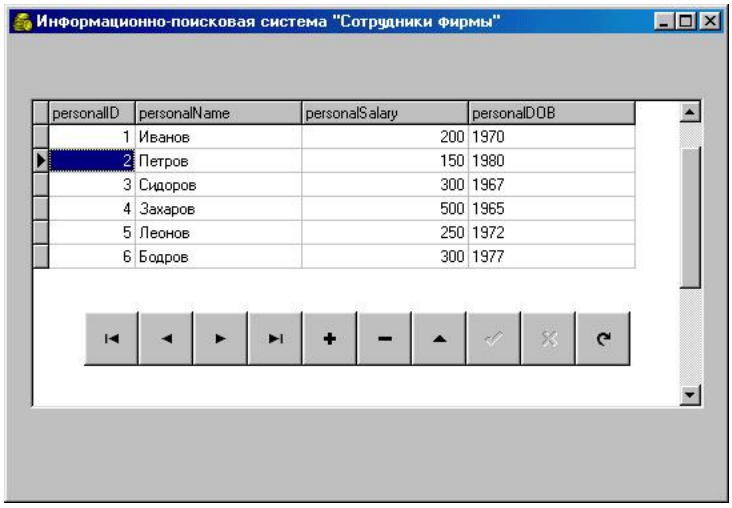


Рис.19

17. Используя компоненты для работы с БД (закладки/страницы палитры компонентов: **Data Eccess**, **Data Controls**, **BDE**, **ADO**), постарайтесь самостоятельно организовать такие операции с таблицами БД как редактирование данных, а не только их просмотр, установив подсветку текущей строки одновременно по всем полям.

Выводы:

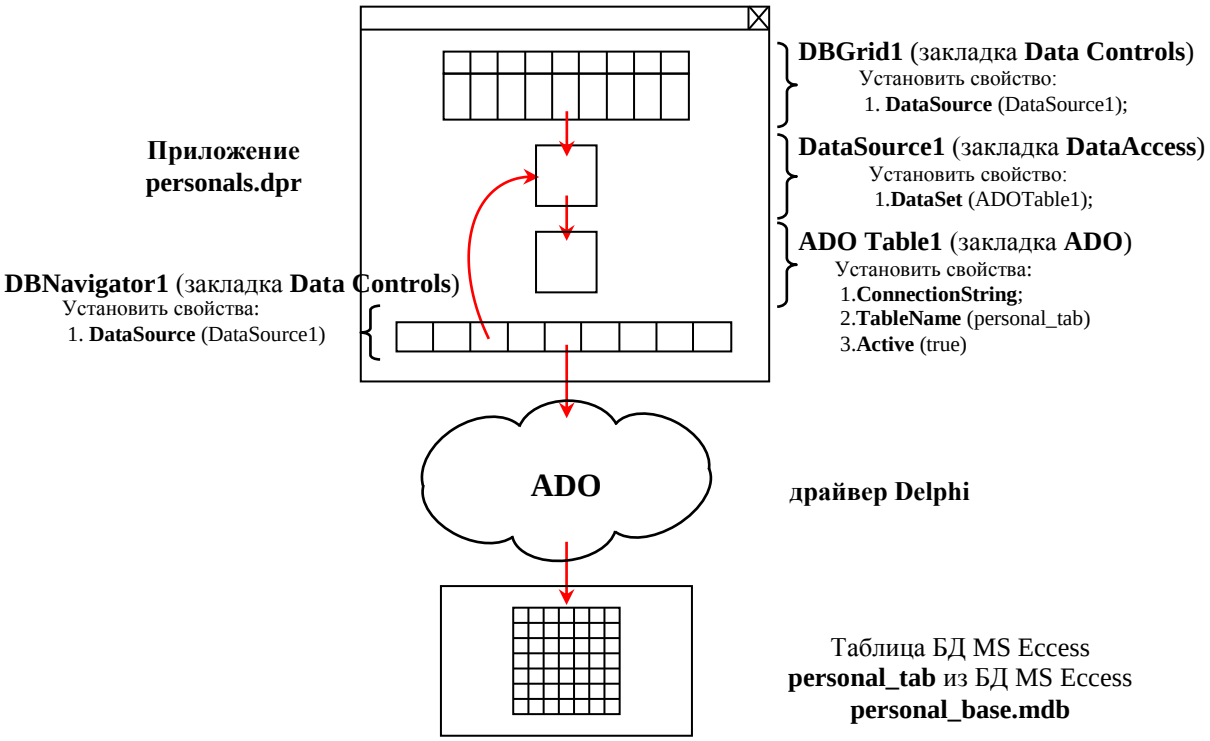
1. Таким образом мы организовали в программной среде Delphi *навигационную* работу с реляционной базой данных ***.mdb**, созданной в MS Eccess.

2. Драйвер **OLE DB** организуется в виде библиотеки функций **OLE DB Libriary**, которая в среде Delphi предоставляет различные функции в виде библиотеки **ADO** – **A**ctiveX **D**ata **O**bjects (объекты данных ActiveX), представляющие собой совокупность драйверов, упакованных в библиотеку классов в виде определенных компонентов.

3. Особенность организации навигационной работы с базой данных в среде Delphi состоит в том, что фирма Borland разделила функции в библиотеке ADO на две:

- связь с таблицей (компонент ADO Table в закладке ADO);
- открытие таблицы (компонент DBGrid в закладке Data Controls).

В связи с этим устанавливаемые между компонентами связи имеют вид:



где: ↓ - обозначение связей между компонентами Delphi - ссылка объектов друг на друга в виде, которые в общем случае бывают в виде:

1) ***ассоциаций***, которые могут быть однонапрвленными и двунаправленными;
2) ***агрегаций*** – один объект находится внутри другого

Рис. 20

3.2.3. Основные изменения в программном коде после разработки интерфейса клиентского приложения (выделены жирным шрифтом)

```
 unit firmpersonals;
interface                          // Раздел объявлений
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids;
type
  TForm1 = class(TForm)
    ADOTable1: TADOTable
    DBGrid1: TDBGrid
    DataSource1: TDataSource
    DBNavigator1: TDBNavigator
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
Implementation                     // Раздел реализации
{$R *.dfm}
end.
```

```
program personals;
uses
  Forms,
  firmpersonals in 'firmpersonals.pas' {fmMain};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TfmMain, fmMain);
  Application.Run;
end.
```

ЛИТЕРАТУРА

1. Федотова С.В. Создание Windows-приложений в среде Delphi [Электронный ресурс]/ Федотова С.В. – Электрон. текстовые данные. – М.: СОЛОН-ПРЕСС, 2010. – 220 с. – Режим доступа: http://www.iprbookshop.ru/8664. – ЭБС «IPRbooks».
2. Белов В.В. Программирование в Delphi. Процедурное, объектно-ориентированное, визуальное [Электронный ресурс]: учебное пособие для вузов/ Белов В.В., Чистякова В.И. – Электрон. текстовые данные. – М.: Горячая линия – Телеком, 2014. – 240 с. – Режим доступа: http://www.iprbookshop.ru/37133. – ЭБС «IPRbooks».
3. СУБД. Язык SQL в примерах и задачах [Электронный ресурс]: учебное пособие/ И.Ф. Астахова [и др.]. – Электрон. текстовые данные. – М.: ФИЗМАТЛИТ, 2009. – 168 с. – Режим доступа: http://www.iprbookshop.ru/12971. – ЭБС «IPRbooks».
4. Когаловский М.Р. Перспективные технологии информационных систем [Электронный ресурс]/ Когаловский М.Р. – Электрон. текстовые данные. – М.: ДМК Пресс, 2009. – 287 с. – Режим доступа: http://www.iprbookshop.ru/7637. – ЭБС «IPRbooks».
5. Роббинс Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. М.: Эксмо, 2014. – 528 с.
6. Р.Райордан. Основы реляционных баз данных – М.: Русская Редакция, 2001. – 384 с.

Составитель: Маргарита Петровна Лапчинская

М Е Т О Д И Ч Е С К И Е У К А З А Н И Я

по дисциплине «Методы и средства проектирования ИС и технологий»

Расчетно-графическая работа №3

«Разработка интерфейса информационной системы справочного назначения в среде Delphi. Навигационный доступ к данным»

Подп. к печ.	**Формат бумаги 60x90**	**Бумага офсетная**	**Печ.л. 0,83**
Уч.-изд.л. 0,83	Тираж 5 экз.	Заказ №	Цена договорная

МИИГАиК

105064, Москва К-64, Гороховский пер., 4