



Автономная некоммерческая образовательная организация
высшего образования Центросоюза Российской Федерации
«Сибирский университет потребительской кооперации»

Методические указания
и задания по выполнению лабораторных занятий
и самостоятельных работ
по дисциплине

ОП.09 ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

по специальности
**09.02.13 Интеграция решений с применением
технологий искусственного интеллекта**

(направленность программы: Применение искусственного интеллекта)

квалификация выпускника:
Специалист по работе с искусственным интеллектом

Новосибирск
2025

Методические указания и задания к практическим занятиям, самостоятельной работе по дисциплине *«Основы проектирования информационных систем»* для обучающихся по специальности 09.02.13 *Интеграция решений с применением технологий искусственного интеллекта*/ [сост. О.Н. Иванова, канд. экон. наук];– Новосибирск, 2025.

РЕЦЕНЗЕНТ: Колдунова И.Д., канд. пед. наук, ст. преподаватель кафедры информатики

Методические указания и задания к практическим занятиям, самостоятельной работе по дисциплине *«Основы проектирования информационных систем»* рекомендованы к использованию в учебном процессе на заседании кафедры информатики, протокол от 28 мая 2025 г., № 9.

.

.

СОДЕРЖАНИЕ

1. Общие положения.....	4
2. Темы и их краткое содержание.....	5
3. Методические указания и задания к лабораторным занятиям и самостоятельной работе	6
5. Вопросы для самостоятельной работы	82
4. Список рекомендуемой литературы	84
6. Перечень информационных ресурсов	84
7. Учебно-методическое обеспечение	85

1. ОБЩИЕ ПОЛОЖЕНИЯ

Методические указания по выполнению лабораторных и самостоятельных работ по дисциплине «ОП.09 Основы проектирования информационных систем» предназначены для реализации программы по специальности 09.02.13 *Интеграция решений с применением технологий искусственного интеллекта* входящей в общепрофессиональный цикл дисциплин.

Методические указания и задания разработаны и составлены на основе рабочей программы ОП.09 Основы проектирования информационных систем. Указанная учебная дисциплина относится к общепрофессиональному циклу в структуре основной профессиональной образовательной программы.

Методические указания и задания предназначены для подготовки и выполнения обучающимися лабораторных работ по дисциплине. Методические указания по выполнению лабораторных и самостоятельных работ по дисциплине «ОП.09 Основы проектирования информационных систем» включают в себя теоретические сведения, практические задания, контрольные вопросы для каждой лабораторной работы.

Выполнение лабораторных работ предназначено для получения умений и навыков работы с СУБД MS Access, Visual FoxPro.

Наличие лабораторных работ позволяет отследить уровень умений и степень подготовленности обучающихся к выполнению профессиональных задач.

2. ТЕМЫ И ИХ КРАТКОЕ СОДЕРЖАНИЕ

Тема 1. Основные понятия баз данных

Предметная область. Описание предметной области. Структурирование. Объект. Класс объектов. Связи между объектами.

Информационная модель данных. Последовательность создания информационной модели.

Понятие нормализации отношений. Первая нормальная форма. Вторая нормальная форма. Функциональная зависимость. Третья нормальная форма. Транзитивная зависимость.

Типы связей: один к одному, один ко многим, многие ко многим. Реляционная модель данных. Основные термины: отношение, кортеж, тип данных, домен, атрибут, таблица, запись, связь, запрос. Проектирование данных. Компоненты и бизнес процессы. Система данных. Задача и система задач. Пример формализованного описания задачи.

Система управления базами данных. Системная среда, удовлетворяющая разнообразным требованиям к данным. Типы систем. Иерархия пользователей. Программное обеспечение баз данных. Аппаратное обеспечение баз данных. Краткая история баз данных. СУБД. Примеры СУБД.

MS Access как реляционная СУБД. Обзор MS Access. Общие представле-

ния. Области применения: в малом бизнесе, в работе по контракту, в крупной корпорации, в качестве персональной СУБД. MS Access и Windows. Архитектура MS Access. Объекты MS Access. Список основных объектов: таблица, запрос, форма, отчет, макрос, модуль. Взаимосвязь основных объектов MS Access. Справочная система. Окно базы данных. Определение данных и их хранение. Обработка данных. Управление данными. Построение базы данных в MS Access. Типичная задача и использование Мастера баз данных. MS Access как средство разработки приложений.

Тема 2. Взаимосвязи в моделях и реляционный подход к построению моделей

Создание новой базы данных. Создание таблиц базы данных. Режимы создания таблиц. Структура таблиц. Свойства полей и таблиц. Типы данных. Ключевое поле. Маска ввода. Создание и изменение связей между таблицами.

Добавление записей в таблицу. Удаление записей из таблицы. Сортировка, поиск и фильтрация данных в таблице. Импорт, экспорт и связывание таблиц.

Режимы создания запросов. Запрос на выборку. Конструктор запросов. Добавление полей в запрос. Задание условия отбора и порядка сортировки результатов запроса. Перекрестные запросы. Запросы на изменение. Запросы с параметрами. Использование операторов и выражений в запросах. Построитель выражений.

Типы и назначение форм. Режимы и этапы создания форм. Макет формы в режиме конструктора. Элементы управления и их свойства. Сложные много-табличные формы.

Сравнительный обзор форм и отчетов. Типы отчетов. Способы и этапы создания отчетов. Макет отчета. Редактирование отчета. Группировка данных в отчете. Вывод на печать.

Тема 3. Этапы проектирования баз данных

Гиперссылки. Редактирование и вставка гиперссылок. Задание гиперссылок к страницам или серверам Intranet. Экспорт таблиц и запросов в HTML. Использование шаблонов HTML. Экспорт отчетов в HTML. Использование мастера публикаций в Web для создания статистической Web-страницы.

Тема 4. Проектирование структур баз данных

OLE-технология. Добавление графических данных в формы и отчеты. Импорт и реорганизация данных рабочего листа. Использование Excel в качестве сервера OLE. Использование средства слияния MS Word с базами данных Access.

Тема 5. Организация запросов SQL

Достоинства и недостатки языка SQL. Стандарты языка SQL. Возможности современного языка SQL. Общая структура директивы SELECT. Однотабличные запросы на выборку данных. Выборка записей по заданному условию. Сортировка результатов запроса. Встроенные функции языка SQL. Запросы с

группировкой записей. Групповые (агрегатные) функции. Использование вложенных запросов (подзапросов). Многотабличные запросы с операцией JOIN. Работа с результатами нескольких запросов. Запросы на изменение данных в таблицах БД (операторы UPDATE и DELETE). Использование оператора MERGE.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ И САМОСТОЯТЕЛЬНОЙ РАБОТЕ

Самостоятельная работа является одним из видов учебной деятельности обучающихся, способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня. Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений обучающихся;
- углубления и расширения теоретических знаний;
- формирования умений использовать специальную литературу;
- развития познавательных способностей и активности обучающихся, творческой инициативы, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации.

Аудиторная самостоятельная работа по учебной дисциплине проводится на учебных занятиях под непосредственным руководством преподавателя и по его заданию (сквозная задача приведена ниже). Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия для обучающихся очной форм обучения.

Основными видами аудиторной самостоятельной работы являются:

- обсуждение теоретических вопросов и решение практических задач по темам дисциплины;
- работа с литературой и другими источниками информации, в том числе электронными.

Решение сквозной задачи осуществляется на лабораторных занятиях в соответствии с графиком учебного процесса. Для подготовки к лабораторным занятиям обучающиеся должны изучить соответствующую главу основного учебного пособия, рекомендованную специальную литературу. Кроме того, необходим самостоятельный поиск и подбор литературы, включая монографии и журнальные публикации, информацию из сети Интернет.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя, но без его непосредственного участия. Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит консультацию с определением цели задания, его содержания, сроков выполнения, ориентировочного объема работы, основных требований к результатам работы, критериев оценки, форм контроля и перечня литературы. В процессе консультации преподаватель

предупреждает о возможных типичных ошибках, встречающихся при выполнении задания.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

- подготовка сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов;
- решение задач.

При выполнении внеаудиторной самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания. Контроль результатов внеаудиторной самостоятельной работы обучающихся может проводиться в письменной, устной или смешанной форме.

Лабораторная работа 1 ***«Нормализация реляционной БД, освоение принципов проектирования БД»***

Тема 3 Этапы проектирования баз данных

Цель работы: «Знакомство реляционной БД, освоение принципов проектирования БД» ***Материально-техническое обеспечение:*** Компьютер, операционная система Windows 7,

MS Access 2010

Краткие теоретические сведения:

Общие сведения о программе MS Access

Access является наиболее сложной программой из всех офисных приложений Microsoft Office. Чтобы начать работу с этой программой, вначале необходимо создать структуру базы данных, подробно ее описать, а затем создать различные формы.

ACCESS – это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных, эти таблицы между собой связаны. Таблицу ACCESS можно связать с данными, хранящимися на другом компьютере. Данные ACCESS очень просто комбинировать с данными EXCEL, WORD и другими программами Office.

Access во многом похож на Excel. Основное различие между таблицей БД и электронной таблицей – в системе адресации: в электронной таблице адресуется каждая ячейка, а в таблице БД – только поля текущей записи. **База данных состоит из следующих компонентов:**

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах. В БД столбцы называются ***полями***, а строки – ***записями***. Для работы таблицу необ-

ходимо открыть. Перед окончанием работы ее следует закрыть, предварительно сохранив все изменения.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули - это программы, написанные на языке программирования Visual Basic.

Таблицы

С таблицами можно работать в двух режимах — **таблицы** и **конструктора**.

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически. Полям присваиваются имена с последовательными номерами («Поле1», «Поле2» и т. д.).

На вкладке **Создание** в группе **Таблицы** щелкните **Таблица**. 

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. При этом будет выделена первая пустая ячейка столбца **Добавить поле..**

1. Чтобы добавить данные, начните их вводить в первую пустую ячейку или вставьте данные из другого

2. Для переименования столбца (поля) дважды щелкните заголовок столбца и введите новое имя. Рекомендуется присваивать полям значимые имена, чтобы при просмотре области **списка полей** было понятно, что содержится в каждом поле.

3. Для перемещения столбца сначала щелкните его заголовок, чтобы выделить столбец, а затем перетащите столбец в нужное место.

4. Кроме того, можно выбрать сразу несколько смежных столбцов, а затем одновременно перетащить их в новое место. Чтобы выбрать несколько последовательно расположенных столбцов, щелкните заголовок первого столбца, а затем, удерживая нажатой клавишу SHIFT, щелкните заголовок последнего столбца.

Создание таблицы в режиме конструктора

В режиме конструктора сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например, вставку или импорт. 1. На вкладке **Создание** в группе **Таблицы** щелкните **Конструктор таблиц**. 

2. Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.

3. Когда все необходимые поля будут добавлены, сохраните таблицу:

4 Щелкните значок **Кнопка Microsoft Office** , а затем выберите команду **Сохранить**.

Создание таблицы на основе шаблона

В приложении Access имеются шаблоны для часто используемых типов таблиц. Одним щелчком мыши можно создать полную, сконфигурированную и готовую к использованию структуру таблицы. Чтобы привести таблицу в соответствие со своими запросами пользователь может добавлять или удалять поля.

1. На вкладке **Создание** в группе **Таблицы** щелкните **Шаблоны таблиц** и затем выберите из списка один из доступных шаблонов.

2. Данные можно ввести в первую пустую ячейку таблицы или вставить их из другого источника в соответствии с инструкцией.

Удаление столбца

Щелкните заголовок столбца правой кнопкой мыши, а затем выберите команду

Удалить столбец .

Свойства полей. Типы полей

Поля – это основные элементы структуры базы данных, которые обладают свойствами. Основным свойством любого поля является его длина, которая выражается в символах. Уникальным свойством любого поля является его *Имя*.

Все записи состоят из полей, в каждом из которых содержится отдельный фрагмент информации. При создании таблицы необходимо сначала указать имена и типы полей, а затем вводить данные

Разные типы полей имеют разное назначение и разные свойства

Типы данных	Описание
Текстовый	Произвольный набор(не более 255) букв, цифр, знаков пунктуации. Может использоваться для хранения имен, адресов, телефонов, кратких описаний.
Числовой	Используется для хранения чисел
Дата/Время	Для хранения информации о дате и времени до 9999 года включительно
Денежный	Денежные значения и числовые данные, используемые в математических расчетах
Поле МЕМО	Для хранения комментариев до (64Кб)

Счетчик	Специальное числовое поле, в котором Access автоматически присваивает уникальный порядковый номер каждой записи.. Значение полей типа счетчика обновлять нельзя
Поле объекта OLE	Позволяет поместить в таблицу произвольный объект из другой программы
Гиперссылка	Адрес в Интернете, электронный адрес или адрес файла
Мастер подстановок	Создает поле, в котором предлагается выбор значений из списка или из поля со списком, содержащего набор постоянных значений или значений из другой таблицы
Логический	Содержит одно из двух значений: Истина или ЛОЖЬ

Правильная структура базы данных

В основе процесса создания базы данных лежат определенные принципы. Первый принцип заключается в необходимости исключать повторяющиеся (или лишние) данные.

Второй принцип касается важной роли правильных и полных данных. Если база данных содержит неправильные данные, все отчеты, в которых данные объединяются, будут также содержать неверные сведения. Это может привести к принятию неправильных решений на основе отчетов.

Процесс разработки

Процесс разработки базы данных включает следующие шаги: Определение цели создания базы данных, поиск и организация необходимых данных, распределение данных по таблицам, преобразование элементов данных в столбцы **Задание первичных ключей.**

Каждая таблица должна содержать столбец или набор столбцов для однозначного определения каждой строки таблицы. Как правило, в этих целях используется уникальный идентификационный номер, например код сотрудника или серийный номер. В базе данных такие сведения носят название **первичного ключа** таблицы.

Создание связей между таблицами

Проанализируйте все таблицы и определите, как данные одной таблицы связаны с данными других таблиц. Добавьте в таблицы поля или создайте новые таблицы для создания необходимых связей.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

СОЗДАНИЕ БАЗОВЫХ ТАБЛИЦ

Руководитель малого предприятия, выполняющего сборку персональных компьютеров из готовых компонентов, заказал разработку базы данных, основанной на двух таблицах комплектующих. Одна таблица содержит данные, которые могут отображаться для клиентов при согласовании спецификации изделия, - в ней указаны розничные цены на компоненты. Вторая таблица предназначена для анализа результатов деятельности предприятия - в ней содержатся оптовые цены на компоненты и краткая информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы):

1. Запустите программу Microsoft Access (*Пуск → Программы → Microsoft Access*).

2. В окне *Microsoft Access* включите переключатель *Новая база данных* и щелкните на кнопке *ОК*.

3. В окне *Файл новой базы данных* выберите свой диск *U*, создайте новую папку с именем *База данных* и в нее сохраните файл, дав ему имя: *Комплектующие*. Убедитесь, что в качестве типа файла выбрана база данных *Microsoft Access*, и щелкните на кнопке *Создать*. Откроется окно новой базы - *Комплектующие: база данных*.

4. Откройте панель *Таблицы*.

5. Дважды щелкните на значке *Создание таблицы* в режиме конструктора - откроется бланк создания структуры таблицы.

6. Для первой таблицы введите следующие поля:

Имя поля	Тип данных	Описание
▶ Компонент	Текстовый	
Модель	Текстовый	
Основной параметр	Текстовый	
Цена	Денежный	

7. Щелкните на поле *Цена*. В нижней части бланка задайте свойство *Число десятичных знаков* равным 2.

8. Для связи с будущей таблицей поставщиков надо задать ключевое поле. Поскольку здесь ни одно поле явно не претендует на "уникальность", используем комбинацию полей *Компонент* и *Модель*. Выделите оба поля в верхней части бланка (при нажатой клавише *SHIFT*). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт *Ключевое поле*.

9. Закройте окно *Конструктора*. При закрытии окна дайте таблице имя *Комплектующие*.

10. Повторив действия пп. 5...9, создайте таблицу *Поставщики*, в которую входят следующие поля.

Поставщики : таблица		
	Имя поля	Тип данных
🔑	Компонент	Текстовый
🔑	Модель	Текстовый
	Цена оптовая	Денежный
	Поставщик	Текстовый
	Телефон	Текстовый
	Адрес	Текстовый
	Примечание	Поле МЕМО

Обратите внимание на то, что поле номера телефона является текстовым, несмотря на то, что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают по величине, не вычитают из одного номера другой и т.д. Это типичное текстовое поле.

! Не забудьте задать ключевые поля в таблице «Поставщики» - это поле «Компонент» и поле «Модель»

СОЗДАНИЕ МЕЖТАБЛИЧНЫХ СВЯЗЕЙ

1. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Убедитесь, что на ней присутствуют значки ранее созданных таблиц *Комплектующие* и *Поставщики*.

2. Разыщите на панели инструментов кнопку *Схема данных*. Если есть сложности, найдите команду строки меню: *Сервис* → *Схема данных*. Воспользуйтесь любым из этих средств, чтобы открыть окно *Схема данных*. Одновременно с открытием этого окна открывается диалоговое окно *Добавление таблицы*, на вкладке *Таблицы* которого можно выбрать таблицы, между которыми создаются связи.

3. Щелчком на кнопке *Добавить* выберите таблицы *Комплектующие* и *Поставщики*, в окне *Схема данных* откроются списки полей этих таблиц.

4. При нажатой клавише *SHIFT* выделите в таблице *Комплектующие* два поля - *Компонент* и *Модель*.

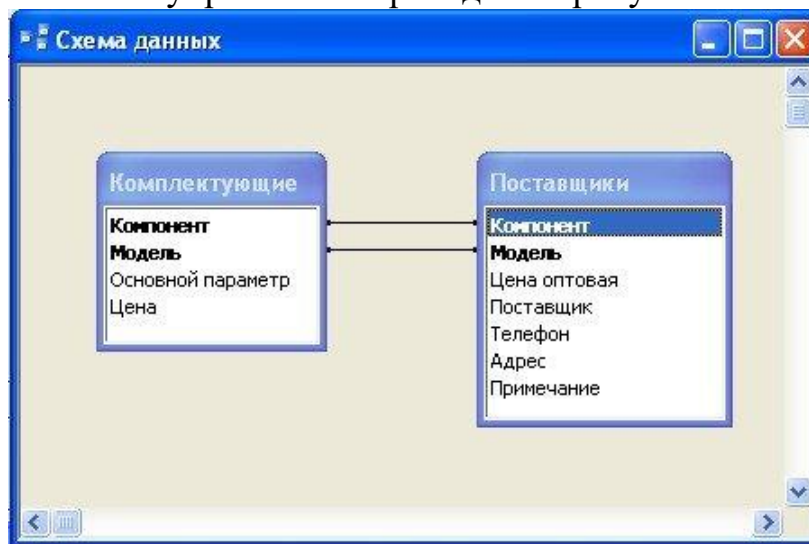
5. Перетащите эти поля на список полей таблицы *Поставщики*. При отпуске кнопки мыши автоматически откроется диалоговое окно *Схема связей*.

6. На правой панели окна *Изменение связей* выберите поля *Компонент* и *Модель* таблицы *Поставщики*, включаемые в связь. Не устанавливайте фла-

жок *Обеспечение целостности данных* - в данном упражнении это не требуется, но может препятствовать постановке учебных опытов с таблицами.

9. Закройте диалоговое окно *Изменение связей* и в окне *Схема данных* рассмотрите образовавшуюся связь. Убедитесь в том, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой кнопки мыши откроется контекстное меню, позволяющее разрывать связь или редактировать ее.

Результат выполнения упражнения приведен на рисунке.



СОЗДАНИЕ ЭКРАННОЙ ФОРМЫ

Наполнять базу данных можно двумя способами:

- Непосредственно вводя информацию в таблицы БД;
- Вводя информацию через специально созданные экранные формы.

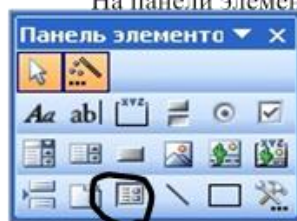
Экранная форма должна обеспечивать однократный ввод данных во взаимосвязанные таблицы.

Создайте с помощью мастера многотабличную форму, состоящую из двух форм (главной и подчиненной). В качестве источника данных для основной части формы указать таблицу **Комплектующие**, для подчиненной формы - таблицу **Поставщики**.

Таблица **Комплектующие** является главной по отношению к таблице **Поставщики**.

1. Создайте форму в режиме мастера на основе полей таблицы **Комплектующие**.
2. Затем откройте созданную форму в режиме конструктора

На панели элементов найдите кнопку Подчиненная форма/отче



Нажав кнопку щелкните левой кнопкой мыши по Примечанию формы **Комплектующие** и создайте подчиненную форму на основе имеющейся таблицы **Поставщики**.

Созданная форма будет выглядеть следующим образом:

Данные для заполнения таблицы **Комплектующие**.

Компо- нент	Модель	Основной параметр	Це- на
Винче- стер HDD	Seagate ST902503FJD1 05-RK	250 Gb	5903 р.

Винчестер HDD	Hitachi SATA-II HDT721016SL A380	160 Gb	1618 p.
Винчестер HDD	Maxtor SATA-II STM3160813AS	160 Gb	1624 p.
Материнская плата	Socket AM2+ ASUS	M3N78 GF8200 DDR2 SVGA PCI-Ex16 Sound GLAN ATX	3058 p.
Материнская плата	Socket AM2 Gigabyte	GA-M56S-S3 nForce560 DDR2 PCI-E 8ch GLAN 1394 ATX	2556 p.
Материнская плата	Socket AM2 MSI	K9N NEO-F nVidia nForce 550 DDR2 SATA-II Audio, GLAN ATX	2313 p.
Корпус	Miditower Foxconn TH-002 Silver-Black 500Wt ATX (P4) гориз.БП		4027 p.
Корпус	Gigabyte	GZ-X5BPD black w/o PSU ATX	2592
Корпус	Miditower GMC	C-50 350W Black, SlimATX 2.0 (serialATA, PCI-E, fan 80mm) 4xUSB2.0	2552
Видеокарта	PCI-E 512 Mb ASUS	EAH4650DI DDR2 2DVI TV	2940
Видеокарта	PCI-E 512 Mb Sapphire	HD4830 DDR3 Dual DVI	4735
Видеокарта	PCI-E 1 Gb MSI	N9600GT-T2D1G-OC GF9600GT DDR3 TV DVI	5687
Процессор	Pentium Dual-Core	E5400 (2,7 GHz 2Mb 800 MHz EM64T) Socket 775	3995
Процессор	Celeron Dual Core	E1200 (1,6 ГГц/ 512Kb 800MHz Socket 775 / EM64T)	1874

Процес- сор	Athlon 64 X2 Dual Core	4800+ Socket AM2 AMD	2121
Накопи- тель DVD±R W	NEC	AD-7201A Silver/DVD±RW/DL ±12x/DLRW +2.4 -2x/DVD-RAM 12x/±20xW/+8x-6x RW	1085
Накопи- тель DVD±R W	ASUS	DRW-20B1LT SATA Silver DVD 20x/6x/8x/16x, CD 48x/32x/48x	1241
Накопи- тель DVD±R W	NEC	AD-7200A IDE Silver 8xDL/20xDVD±R/8xDVD+RW/6xDVDRW/5x RAM/16xDVD-ROM	1015
Монитор	ViewSonic VA903B"	TFT 19"	5951
Монитор	Benq G900AD	TFT 19"	5951
Монитор	Samsung 2023NW	TFT 20"	6390
Клавиа- тура	BTC	5213 USB и-нет, м-медиа, эргономичная	385
Клавиа- тура	Genius KB06X	с подставкой под запястья (Palm Rest) black PS/2	303
Клавиа- тура	Genius KB06XE	Black PS/2	232
Мышь	BTC M843U- Grey	оптика, 1000 dpi, серая, USB	249
Мышь	Genius Traveler 355 Laser	оптический скролл, mini, лазерная, 1600/800dpi, USB, black	834
Мышь	Genius NetScroll 100 Optica	(Зкн. и scroll) PS/2 silver	156

Данные для заполнения информации о поставщиках придумайте свои, цены оптовые сделайте ниже цен реализации.

Контрольные вопросы:

1. Что такое СУБД?
2. Что такое таблица в БД?
3. Назовите основные элементы БД в MS Access.

Лабораторная работа 2

«Проектирование реляционной БД. Нормализация таблиц.».

Тема 3 Этапы проектирования баз данных

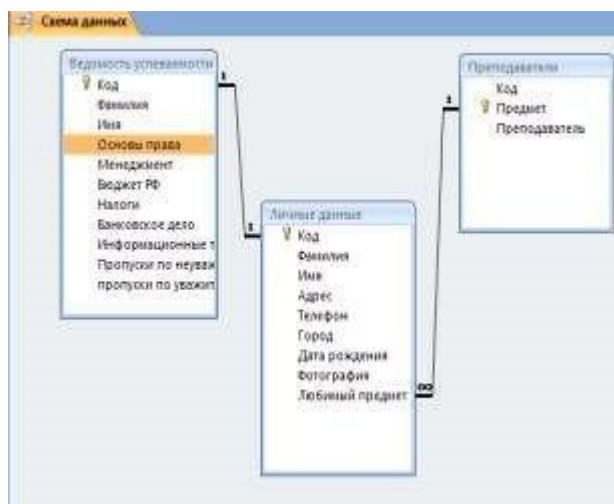
Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, MS Access 2010

Краткие теоретические сведения:

Связи между таблицами

Для полноценной работы Базы данных нужно наладить связи между таблицами, сделать их зависимыми друг от друга, чтобы по данным одной таблицы можно было узнать данные другой таблицы.



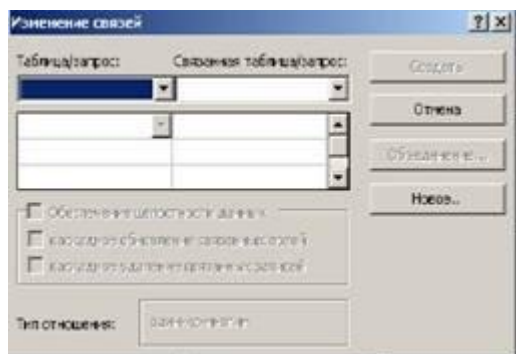
Связь между таблицами осуществляется с помощью числового идентификатора. Взяв идентификатор из одной таблицы, можно найти данные в другой таблице. Для установки связи между двумя таблицами необходимо в группе. **Показать или скрыть**, расположенной на вкладке **Работа с базами данных**,

выбрать команду **Схема данных**. Появится окно **Схема данных**, в котором отображаются зависимости между таблицами.

Добавить новую связь можно, дважды щелкнув левой кнопкой мыши по окну или выбрав соответствующую команду

на вкладке **Работа с базами данных** → **Схема данных** → **Конструктор** → **Изменить связь**. Появится окно **Изменение связей**, в котором нужно указать таблицы между которыми устанавливаются связи.

Как же влияет установление связи в таблицах на работу Microsoft Access?



Программ контролирует ввод данных в связанные таблицы по следующим правилам:

- запрещено добавлять значение, отсутствующее в исходной таблице, в связанную таблицу. То есть, запрещен ввод нового любимого предмета в таблицу **Личные данные**, если в таблице **Преподаватели** этот предмет отсутствует;
- запрещено удаление в исходной таблице записей, на которые есть ссылка в связанной таблице;
- при изменении записей в исходной таблице меняются записи в связанной таблице.

Связанные записи можно удалять только через исходную таблицу.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

СОЗДАНИЕ ЗАПРОСА НА ВЫБОРКУ

В этом упражнении мы создадим запрос на выборку жестких дисков, имевших емкость не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.

3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно *Добавление таблицы*.

4. В окне *Добавление таблицы* выберите таблицу *Поставщики* и щелкните на кнопке *Добавить*. Закройте окно *Добавление таблицы*.

5. В списке полей таблицы *Поставщики* выберите поля, включаемые в результирующую таблицу: *Компонент*, *Модель*, *Цена оптовая*, *Поставщик*, *Телефон*. Выбор производите двойными щелчками на именах полей.

6. Задайте условие отбора для поля *Компонент*, в соответствующую строку введите *Жесткий диск*. Из таблицы будут выбираться не все изделия, а только жесткие диски.

7. Задайте условие отбора для поля *Цена оптовая*. В соответствующую строку введите < 150 . Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.

8. Нам еще надо задать условие отбора по основному потребительскому параметру - емкости жесткого диска. Однако в таблице *Поставщики* такого поля нет. С другой стороны, в ней есть поле *Модель*, которое однозначно определяет параметры изделия. Благодаря тому, что по полю *Модель* у нас установлена связь с таблицей *Комплектующие*, мы получаем возможность ввести в запрос поле *Основной параметр*, взяв его из другой таблицы.

Добавьте список полей таблицы *Комплектующие* в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт *Добавить таблицу* - откроется уже знакомое нам окно *Добавление таблицы*. Выберите в нем таблицу *Комплектующие*.

9. Двойным щелчком на поле *Основной параметр* в списке полей таблицы *Комплектующие* введите это поле в бланк запроса по образцу.

10. В строке *Условие отбора* столбца *Основной параметр* введите условие >8 (емкость диска - более 8 Гбайт).

11. Закройте бланк запроса по образцу. При закрытии запроса введите его имя - *Выбор комплектующих*.

12. В окне *Комплектующие: база данных* откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы *Комплектующие* и *Поставщики* при их наполнении в упражнении 1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

13. По окончании исследований закройте все открытые объекты и завершите работу с программой Microsoft Access.

СОЗДАНИЕ ЗАПРОСОВ "С ПАРАМЕТРОМ"

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 140 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами "с параметром", позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этот прием обеспечивает гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. Запустите программу Microsoft Access - 2010 (*Пуск → Программы → Microsoft Access*).
2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу.
4. Согласно упражнению 3, создайте запрос на выборку, основанный на таблице *Поставщики*, в который войдут следующие поля:
 - *Компонент*;
 - *Модель*;
 - *Цена оптовая*;
 - *Поставщик*;
 - *Телефон*.
5. В строке *Условие отбора* поля *Компонент* введите *Процессор*.
6. Строку *Условие отбора* для поля *Цена оптовая* надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение.

Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 условных единиц, мы бы написали >100 . Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали < 80 . Но если мы хотим дать пользователю возможность выбора, мы должны написать $< [$ Введите максимальную цену $]$.

7. Закройте запрос. При закрытии сохраните его под именем *Выбор комплектующих*.
8. В окне *Комплектующие: база данных* откройте панель *Запросы* и запустите запрос *Выбор комплектующих* - на экране появится диалоговое окно *Введите значение параметра*.
9. Введите какое-либо число и щелкните на кнопке *ОК*. В зависимости от того, что реально содержится в таблице *Поставщики*, по результатам запроса будет сформирована результирующая таблица.
10. Закройте все объекты базы данных. Закройте программу Microsoft Access. **Вывод.** Мы научились формировать запросы "с параметром" и узнали, что в основе этого вида запросов лежат запросы на выборку, у которых в поле *Условие отбора* записан заключенный в квадратные скобки текст, обращенный к пользователю.

СОЗДАНИЕ ИТОГОВОГО ЗАПРОСА

Если полностью заполнить данными таблицу *Комплектующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплектующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *итоговыми запросами*. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле или выполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: "Элитный", "Деловой" и "Экономичный". Несмотря на то, что архитектура всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача - подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

1. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).

2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.

3. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Выберите таблицу *Комплектующие*.

4. Щелчком на значке *Конструктор* откройте таблицу в режиме проектирования - нам это необходимо для создания дополнительного поля *Класс*, в

котором будут храниться данные о том, для какого класса изделий предназначены компоненты.

5. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле *Компонент* и нажмите клавишу *INSERT*.

6. Введите имя нового поля - *Класс* - и определите его тип - *Текстовый*.

7. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.

8. Откройте таблицу *Комплектующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:

- материнская плата;
- процессор;
- оперативная память;
- жесткий диск;
- корпус;
- дисковод CD-ROM;
- дисковод гибких дисков;
- видеоадаптер;
- звуковая карта; • клавиатура; • мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять - в формировании итогового запроса они участвовать не будут.

9. Закройте таблицу *Комплектующие*.

10. Откройте панель *Запросы* щелчком на одноименной кнопке окна *Комплектующие*:
база данных.

11. Выполните двойной щелчок на значке *Создание запроса в режиме конструктора*. В открывшемся диалоговом окне *Добавление таблицы* выберите таблицу *Комплектующие*, на основе которой будет разрабатываться итоговый запрос. Закройте окно *Добавление таблицы*.

12. В бланк запроса по образцу введите следующие поля таблицы *Комплектующие*: *Класс*, *Компонент*, *Цена*.

13. Для поля *Класс* включите сортировку по возрастанию. Включите также сортировку по полю *Цена*, но на этот раз - по убыванию.

14. На панели инструментов Microsoft Access щелкните на кнопке *Групповые операции* или воспользуйтесь строкой меню (*Вид* → *Групповые операции*). Эта команда необходима для создания в нижней части бланка строки *Групповые операции*. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение *Группировка*.

15. Для поля, по которому производится группировка записей (в нашем случае - *Класс*), оставьте в строке *Групповые операции* значение *Группировка*. Для остальных полей щелкните на этой строке - появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

16. Для поля *Цена* выберите итоговую функцию *Sum* для определения стоимости изделия как суммы стоимостей комплектующих.

17. Для поля *Компонент* выберите итоговую функцию *Count*, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

18. Закройте бланк запроса по образцу и дайте ему имя *Расчет стоимости изделия*. Запустите запрос и убедитесь, что он правильно работает.

19. Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

Контрольные вопросы:

1. Для чего необходима связь в таблицах?
2. Как создать запрос на выборку данных?
3. Как создать запрос с «параметром»? 4. Как создать итоговый запрос?

Лабораторная работа 3

«Преобразование реляционной БД в сущности и связи.»

Тема 3 Этапы проектирования баз данных

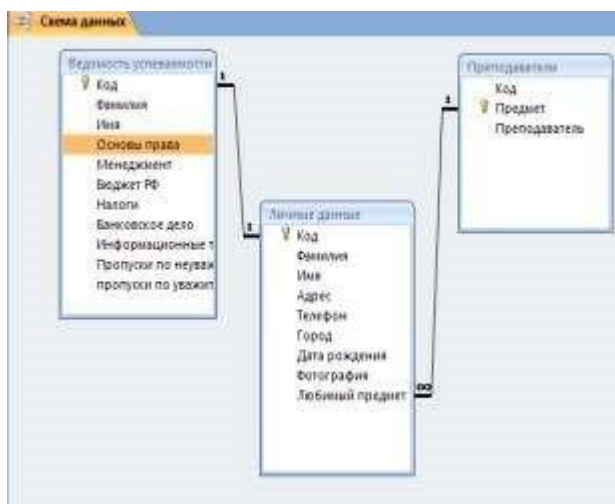
Цель работы: «Проектирование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7,
MS Access 2010

Краткие теоретические сведения:

Связи между таблицами

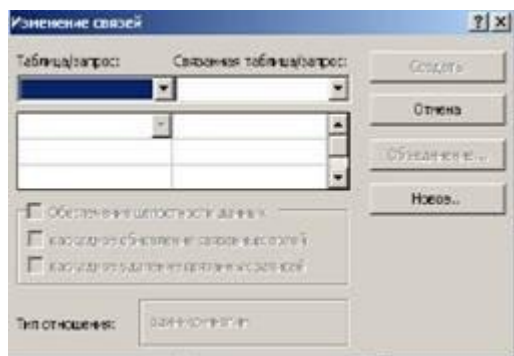
Для полноценной работы Базы данных нужно наладить связи между таблицами, сделать их зависимыми друг от друга, чтобы по данным одной таблицы можно было узнать данные другой таблицы.



Связь между таблицами осуществляется с помощью числового идентификатора. Взяв идентификатор из одной таблицы, можно найти данные в другой таблице. Для установки связи между двумя таблицами необходимо в группе **Показать или скрыть**, расположенной на вкладке **Работа с базами данных**, выбрать команду **Схема данных**. Появится окно **Схема данных**, в котором отображаются зависимости между таблицами.

Добавить новую связь можно, дважды щелкнув левой кнопкой мыши по окну или выбрав соответствующую команду на вкладке **Работа с базами данных** → **Схема данных** → **Конструктор** → **Изменить связь**. Появится окно **Изменение связей**, в котором нужно указать таблицы между которыми устанавливаются связи.

Как же влияет установление связи в таблицах на работу Microsoft Access?



Программ контролирует ввод данных в связанные таблицы по следующим правилам:

- запрещено добавлять значение, отсутствующее в исходной таблице, в связанную таблицу. То есть, запрещен ввод нового любимого предмета в таблицу **Личные данные**, если в таблице **Преподаватели** этот предмет отсутствует;
- запрещено удаление в исходной таблице записей, на которые есть ссылка в связанной таблице;
- при изменении записей в исходной таблице меняются записи в связанной таблице.

Связанные записи можно удалять только через исходную таблицу.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

СОЗДАНИЕ ЗАПРОСА НА ВЫБОРКУ

В этом упражнении мы создадим запрос на выборку жестких дисков, имевших емкость не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).
2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно *Добавление таблицы*.
4. В окне *Добавление таблицы* выберите таблицу *Поставщики* и щелкните на кнопке *Добавить*. Закройте окно *Добавление таблицы*.

5. В списке полей таблицы *Поставщики* выберите поля, включаемые в результирующую таблицу: *Компонент*, *Модель*, *Цена оптовая*, *Поставщик*, *Телефон*. Выбор производите двойными щелчками на именах полей.

6. Задайте условие отбора для поля *Компонент*, в соответствующую строку введите *Жесткий диск*. Из таблицы будут выбираться не все изделия, а только жесткие диски.

7. Задайте условие отбора для поля *Цена оптовая*. В соответствующую строку введите < 150 . Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.

8. Нам еще надо задать условие отбора по основному потребительскому параметру - емкости жесткого диска. Однако в таблице *Поставщики* такого поля нет. С другой стороны, в ней есть поле *Модель*, которое однозначно определяет параметры изделия. Благодаря тому, что по полю *Модель* у нас установлена связь с таблицей *Комплектующие*, мы получаем возможность ввести в запрос поле *Основной параметр*, взяв его из другой таблицы.

Добавьте список полей таблицы *Комплектующие* в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт *Добавить таблицу* - откроется уже знакомое нам окно *Добавление таблицы*. Выберите в нем таблицу *Комплектующие*.

14. Двойным щелчком на поле *Основной параметр* в списке полей таблицы *Комплектующие* введите это поле в бланк запроса по образцу.

15. В строке *Условие отбора* столбца *Основной параметр* введите условие >8 (емкость диска - более 8 Гбайт).

16. Закройте бланк запроса по образцу. При закрытии запроса введите его имя - *Выбор комплектующих*.

17. В окне *Комплектующие: база данных* откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы *Комплектующие* и *Поставщики* при их наполнении в упражнении 1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

18. По окончании исследований закройте все открытые объекты и завершите работу с программой Microsoft Access.

СОЗДАНИЕ ЗАПРОСОВ "С ПАРАМЕТРОМ"

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена

которых менее 140 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами "с параметром", позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этот прием обеспечивает гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. Запустите программу Microsoft Access - 2010 (*Пуск → Программы → Microsoft Access*).
2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу.
4. Согласно упражнению 3, создайте запрос на выборку, основанный на таблице *Поставщики*, в который войдут следующие поля:
 - *Компонент*;
 - *Модель*;
 - *Цена оптовая*;
 - *Поставщик*;
 - *Телефон*.
7. В строке *Условие отбора* поля *Компонент* введите *Процессор*.
8. Строку *Условие отбора* для поля *Цена оптовая* надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение.

Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 условных единиц, мы бы написали >100 . Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали < 80 . Но если мы хотим дать пользователю возможность выбора, мы должны написать $< [$ Введите максимальную цену $]$.
11. Закройте запрос. При закрытии сохраните его под именем *Выбор комплектующих*.
12. В окне *Комплектующие: база данных* откройте панель *Запросы* и запустите запрос *Выбор комплектующих* - на экране появится диалоговое окно *Введите значение параметра*.
13. Введите какое-либо число и щелкните на кнопке *ОК*. В зависимости от того, что реально содержится в таблице *Поставщики*, по результатам запроса будет сформирована результирующая таблица.

14. Закройте все объекты базы данных. Закройте программу Microsoft Access. **Вывод.** Мы научились формировать запросы "с параметром" и узнали, что в основе этого вида запросов лежат запросы на выборку, у которых в поле *Условие отбора* записан заключенный в квадратные скобки текст, обращенный к пользователю.

СОЗДАНИЕ ИТОГОВОГО ЗАПРОСА

Если полностью заполнить данными таблицу *Комплектующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплектующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *итоговыми запросами*. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле или выполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: "Элитный", "Деловой" и "Экономичный". Несмотря на то, что архитектура всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача - подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

7. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).

8. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.

9. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Выберите таблицу *Комплектующие*.

10. Щелчком на значке *Конструктор* откройте таблицу в режиме проектирования - нам это необходимо для создания дополнительного поля *Класс*, в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.

11. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле *Компонент* и нажмите клавишу *INSERT*.

12. Введите имя нового поля - *Класс* - и определите его тип - *Текстовый*.

9. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.

10. Откройте таблицу *Комплектующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:

- материнская плата;
- процессор;
- оперативная память;
- жесткий диск;
- корпус;
- дисковод CD-ROM;
- дисковод гибких дисков;
- видеоадаптер;
- звуковая карта; • клавиатура; • мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять - в формировании итогового запроса они участвовать не будут.

16. Закройте таблицу *Комплектующие*.

17. Откройте панель *Запросы* щелчком на одноименной кнопке окна *Комплектующие*:
база данных.

18. Выполните двойной щелчок на значке *Создание запроса в режиме конструктора*. В открывшемся диалоговом окне *Добавление таблицы* выберите таблицу *Комплектующие*, на основе которой будет разрабатываться итоговый запрос. Закройте окно *Добавление таблицы*.

19. В бланк запроса по образцу введите следующие поля таблицы *Комплектующие*: *Класс*, *Компонент*, *Цена*.

20. Для поля *Класс* включите сортировку по возрастанию. Включите также сортировку по полю *Цена*, но на этот раз - по убыванию.

21. На панели инструментов Microsoft Access щелкните на кнопке *Групповые операции* или воспользуйтесь строкой меню (*Вид* → *Групповые операции*). Эта команда необходима для создания в нижней части бланка строки *Групповые операции*. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение *Группировка*.

22. Для поля, по которому производится группировка записей (в нашем случае - *Класс*), оставьте в строке *Групповые операции* значение *Группировка*. Для остальных полей щелкните на этой строке - появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

20. Для поля *Цена* выберите итоговую функцию *Sum* для определения стоимости изделия как суммы стоимостей комплектующих.

21. Для поля *Компонент* выберите итоговую функцию *Count*, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

22. Закройте бланк запроса по образцу и дайте ему имя *Расчет стоимости изделия*. Запустите запрос и убедитесь, что он правильно работает.

23. Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

Контрольные вопросы:

1. Для чего необходима связь в таблицах?
2. Как создать запрос на выборку данных?
3. Как создать запрос с «параметром»? 4. Как создать итоговый запрос?

Лабораторная работа 4

«Задание ключей. Создание основных объектов БД»

Тема 4 Проектирование структур баз данных

Цель работы: «Задание ключей, создание основных объектов БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, MS Access 2010

Краткие теоретические сведения:

База данных состоит из следующих компонентов:

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах. В БД столбцы называются **полями**, а строки – **записями**. Для работы таблицу необходимо открыть. Перед окончанием работы ее следует закрыть, предварительно сохранив все изменения.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули - это программы, написанные на языке программирования Visual Basic.

Таблицы

С таблицами можно работать в двух режимах — **таблицы** и **конструктора**.

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически. Полям присваиваются имена с последовательными номерами («Поле1», «Поле2» и т. д.).

На вкладке **Создание** в группе **Таблицы** щелкните **Таблица**. 

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. При этом будет выделена первая пустая ячейка столбца **Добавить поле..**

5. Чтобы добавить данные, начните их вводить в первую пустую ячейку или вставьте данные из другого

6. Для переименования столбца (поля) дважды щелкните заголовок столбца и введите новое имя. Рекомендуется присваивать полям значимые имена, чтобы при просмотре области **списка полей** было понятно, что содержится в каждом поле.

7. Для перемещения столбца сначала щелкните его заголовок, чтобы выделить столбец, а затем перетащите столбец в нужное место.

8. Кроме того, можно выбрать сразу несколько смежных столбцов, а затем одновременно перетащить их в новое место. Чтобы выбрать несколько последовательно расположенных столбцов, щелкните заголовок первого столбца, а затем, удерживая нажатой клавишу SHIFT, щелкните заголовок последнего столбца.

Создание таблицы в режиме конструктора

В режиме конструктора сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например, вставку или импорт.

4. На вкладке **Создание** в группе **Таблицы** щелкните **Конструктор таблиц**. 

5. Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.

6. Когда все необходимые поля будут добавлены, сохраните таблицу:

4 Щелкните значок **Кнопка Microsoft Office** , а затем выберите команду **Сохранить**.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

ЗАДАНИЕ 1. СОЗДАНИЕ СТРУКТУРЫ БД.

1.1. Запустить программу Microsoft Access для создания новой базы данных.

Для этого следует нажать кнопки сначала **Пуск**, затем **Программы**, среди которых нужно выбрать **Microsoft Access**.

1.2. Создать новую базу данных под названием МОДЕЛИ, а в скобках указать Фамилию ученика.

Для создания базы данных войдите в **Пуск/Программы/Microsoft Access**. После чего откроется диалоговое окно на экране монитора **Создание новой базы данных**. Нужно выбрать пункт **Новая база данных** и нажать **ОК**, в появившемся окне выбрать диск **С:/Мои документы** и в строке **Имя файла** следует ввести слово **Модели(Фамилию ученика)** и затем нажать кнопку **Создать**. После чего открывается **окно базы данных** под названием **Модели(Фамилия ученика)**. В верхней части окна базы данных представлено меню, где перечислены объекты базы данных:

- **Таблицы,**
- **Запросы,**
- **Формы,**
- **Отчеты,**
- **Макросы, - Модули.**

1.3. Создать структуру новой таблицы.

Для создания структуры новой таблицы в **окне базы данных** следует щелкнуть мышью по кнопке **Таблицы**, затем по кнопке **Создать**. После чего Система откроет диалоговое окно, в котором будет предложено воспользоваться одним из следующих способов создания таблицы:

- **Режим таблицы,**
- **Конструктор,**
- **Мастер таблиц,**
- **Импорт таблиц,**
- **Связь с таблицами.**

Нужно выделить способ **Конструктор**, так как будем конструировать свою таблицу, и нажав **Ок**, структура **Таблицы** открывается в режиме конструктора, куда необходимо ввести

- **Имя поля,**
- **Тип данных, - Описание.**

1.4. Ввести в новую таблицу имена полей и их спецификации так, как это показано в приведенной ниже таблице.

ИМЯ ПОЛЯ	ТИП ДАННЫХ	ОПИСАНИЕ
Код модели	Числовой	Код модели по заводскому заказу
Модель	Текстовый	Тип салона
Мощность	Текстовый	Мощность двигателя (квт/л.с.)
Количество дверей	Числовой	Двухдверный или четырехдверный Салон
Коробка передач	Текстовый	Автоматическая или ручная

Обивка	Текстовый	Велюр, кожа, ткань
Другое оснащение	Мето	Дополнительные аксессуары
Изображение	Поле объекта OLE	Рисунок, представляющий модель
Заводская цена	Денежный	Заводская продажная нетто-цена
Транспортные издержки	Денежный	Издержки на доставку
Предпродажные издержки	Денежный	Издержки на предпродажную Подготовку

1.5. Ввести свойства полей новой таблицы так, как это показано ниже (приведены только те свойства полей, значения которых отличаются от стандартных).

<i>Поле</i>	Раз- Мер Поля	Формат Поля	Число деся- тич- ных Знаков	Значе- ние По Умол- чанию	Обя- за- тель- ное Поле	Индексиро- ван- Ное по- ле
Код модели	Длин- ное це- лое	Основ- ной	0		<i>Да</i>	Да(Не допуска- ются совпадения)
Модель	20				Да	Да(совпадения допуск)
Мощность	10					Нет
Количество Дверей	Целое	Основ- ной	0	4		Нет
Коробка Пере- дач	15			Ручная		<i>Нет</i>
Обивка	20					Нет
Другое Оснащение	Мето					
Изображение					Нет	
Заводская Це- на		Денеж- ный	Авто		Да	Нет
Транспортные Издержки		Денеж- ный	Авто		Да	Нет

Предпродаж- ные Издержки		Денеж- ный	Авто	105	Да	<i>Нет</i>
-------------------------------------	--	---------------	------	-----	----	------------

1.6. Установить первичный ключ таблицы.

Для этого необходимо при выделении поля **Код модели** нажать на панели инструментов **Ключевое поле**.

1.7. Сохранить созданную Структуру таблицы под именем Предлагаемые модели и закрыть таблицу.

1.8. Открыть снова таблицу Предлагаемые модели в режиме Конструктор и отредактировать ее:

- ввести новую строку под названием Цвет перед строкой Количество дверей.

Для вставки нового поля перед существующим необходимо выбрать область маркировки поля слева от существующего поля, далее нажать кнопку **Вставить строку** на панели инструментов. В новую строку ввести название имя поля –Цвет, тип данных текстовый, описание – цвет кузова.

- изменить название поля **Мощность** на название **Мощность двигателя**.

-поменять местами поля таблицы Цвет и Количество дверей.

Для того, чтобы поменять местами предлагаемые поля, необходимо перед полем **Цвет** создать пустую строку, как это было указано выше, затем вырезать строку **Количество дверей** и вставить ее в созданную пустую строку, используя панель инструментов.

Контрольные вопросы:

1. Опишите этапы создания таблицы в конструкторе?
2. Какие типы данных в БД вы знаете? 3. Какие параметры форматирования таблиц вы можете применить в БД?

Лабораторная работа 5
«Создание проекта БД. Создание БД. Редактирование и модификация таблиц»

Тема 4 Проектирование структур баз данных

Цель работы: «создание проекта БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, VFoxPro

Краткие теоретические сведения:

Основные типы полей, используемые в Microsoft Visual FoxPro приведены в таблице

Тип поля	Содержимое
Character (Символьный)	Любые символы. Максимальный размер символьного поля – 254 символа
Numeric (Числовой)	Включает цифры от 0 до 9, знак (необязательно), дробную часть (необязательно)
Float (Вещественный)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с обычной точностью
Double (Двойной Вещественный сдвойной точностью)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с двойной точностью
Integer (Числовой, целочисленный)	Целые числа
Data (Дата)	Дата, определяющая день, месяц и год. Ввод в это поле контролируется VisualFoxPro автоматически
DateTime (Дата и время)	Дата и время. По умолчанию формат вводимого значения имеет вид мм/дд/гг чч:мм:сс
Logical (Логический)	Логические данные. Может принимать только .T. – истина, .F. – ложь

Мемо (Текстовое поле произвольной длины)	Примечания, размер которых неограничен. Может содержать любую символьную информацию, размер которой ограничивается лишь объемом доступного дискового пространства. Фактически – это указатель на блок данных в файле с расширением fpt (файлы dbf и fpt одноименны). Для заполнения данного поля нужно дважды щелкнуть мышью по данному полю (или CTRL+Home), после чего в открывшемся окне набрать нужный текст примечаний. Закрыв окно редактирования (CTRL+W), значение поля будет сохранено, а слово мемо в поле базы данных изменится на Мемо.
General (Общие, двоичное поле произвольной длины)	Поле, используется для хранения ссылки на объект. Под объектом понимается любой текстовый файл, звуковой файл, диаграмма, иллюстрация или другой файл, созданный с помощью приложения, поддерживающего технологию OLE. Для заполнения данного поля нужно двойным щелчком мыши на нем открыть окно редактирования. После чего выбрать пункт меню Edit(правка)-insert object(вставить объект) и в
	открытом выбранном приложении создать нужный объект (например, рисунок). Закрыв окно приложения и окно редактирования поля, поле gen будет заменено на Gen, значение будет сохранено
Currency (Денежный)	Поле для представления денежных сумм, используется для хранения чисел с четырьмя знаками после запятой
Memobinary	Двоичный текст произвольной длины. Содержит любые данные поля мемо, которые можно использовать, не изменяя кодовую страницу
Character binary	Двоичный текст длиной до 254 символов. Содержит любые символьные данные, которые используются, без изменения кодовой страницы

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Запустим Microsoft Visual FoxPro

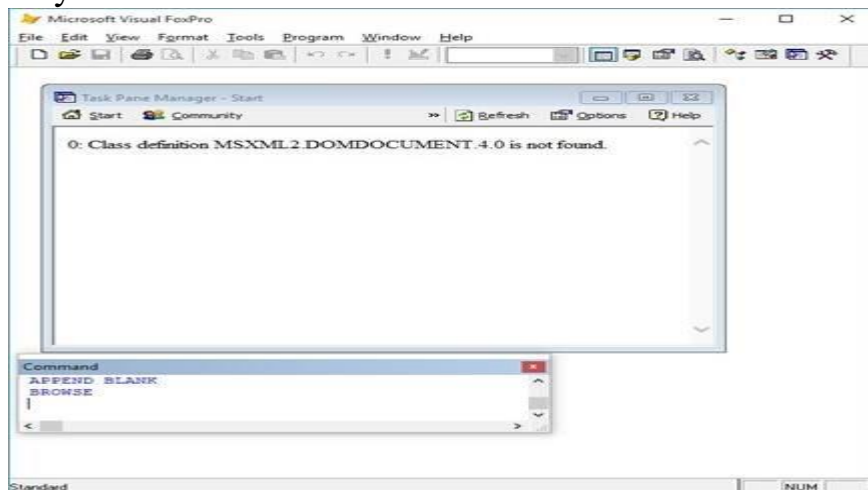


Рисунок 1 У нас появится 2 окна

1. Command – В этом окне мы будем вводить команды
2. Task Pane Manager – окно с инструкцией

Прежде чем начать работать в FoxPro надо ознакомиться с основными командами для создания “Базы данных”

Create <имя> — создать

Modify structure – изменить структуру

Use < имя> — открыть подготовить

Select < имя> — выбрать

Browse – открыть визуализировать

Append blank – добавить запись

Начнём создания нашей первой Базы данных. Для этого напомним Create <имя> в окне команд. У нас откроется окно структуры базы данных

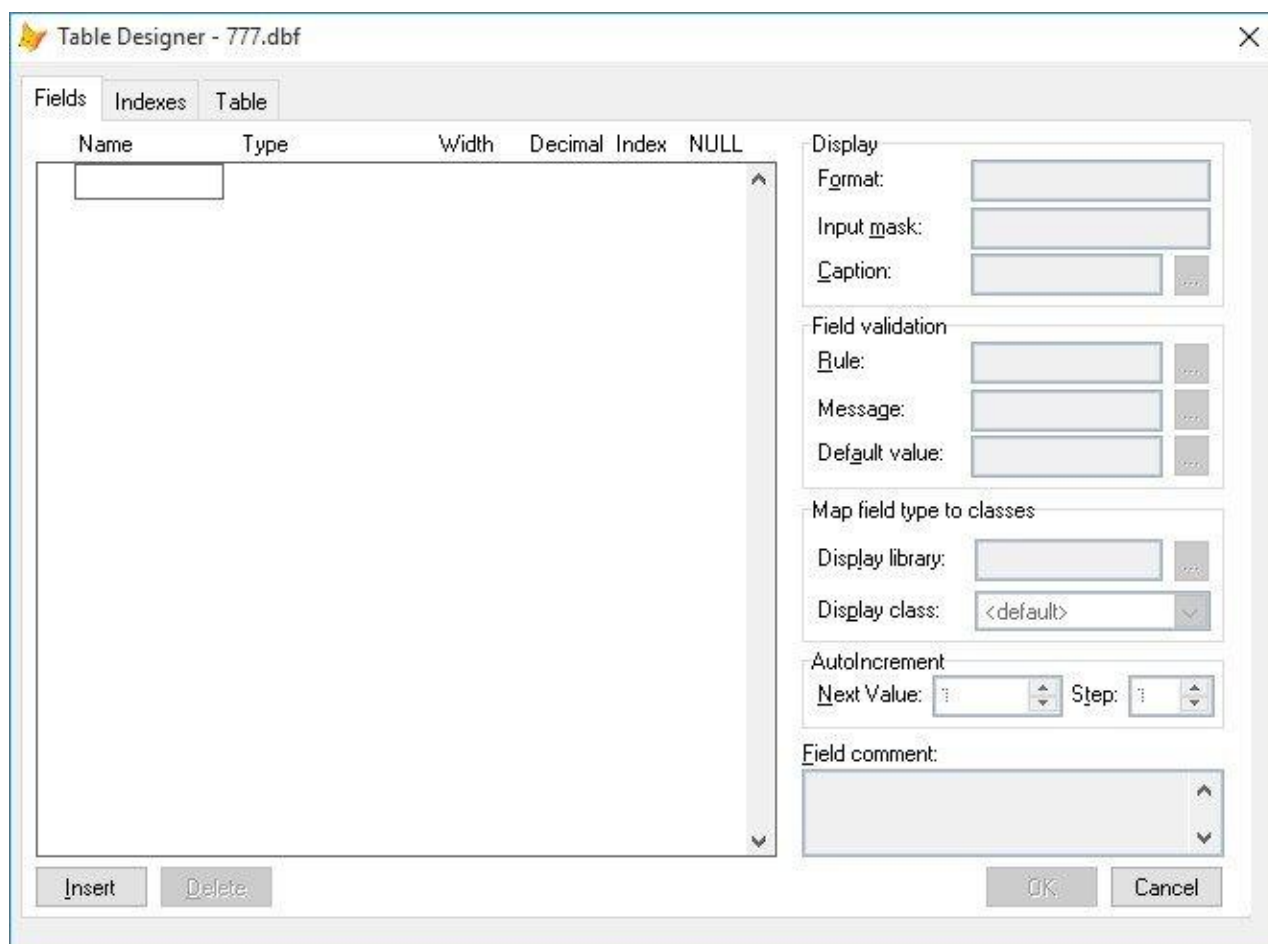


Рисунок 2

Name это название будущего столбца

Type это тип столбца

Width – ширина столбца

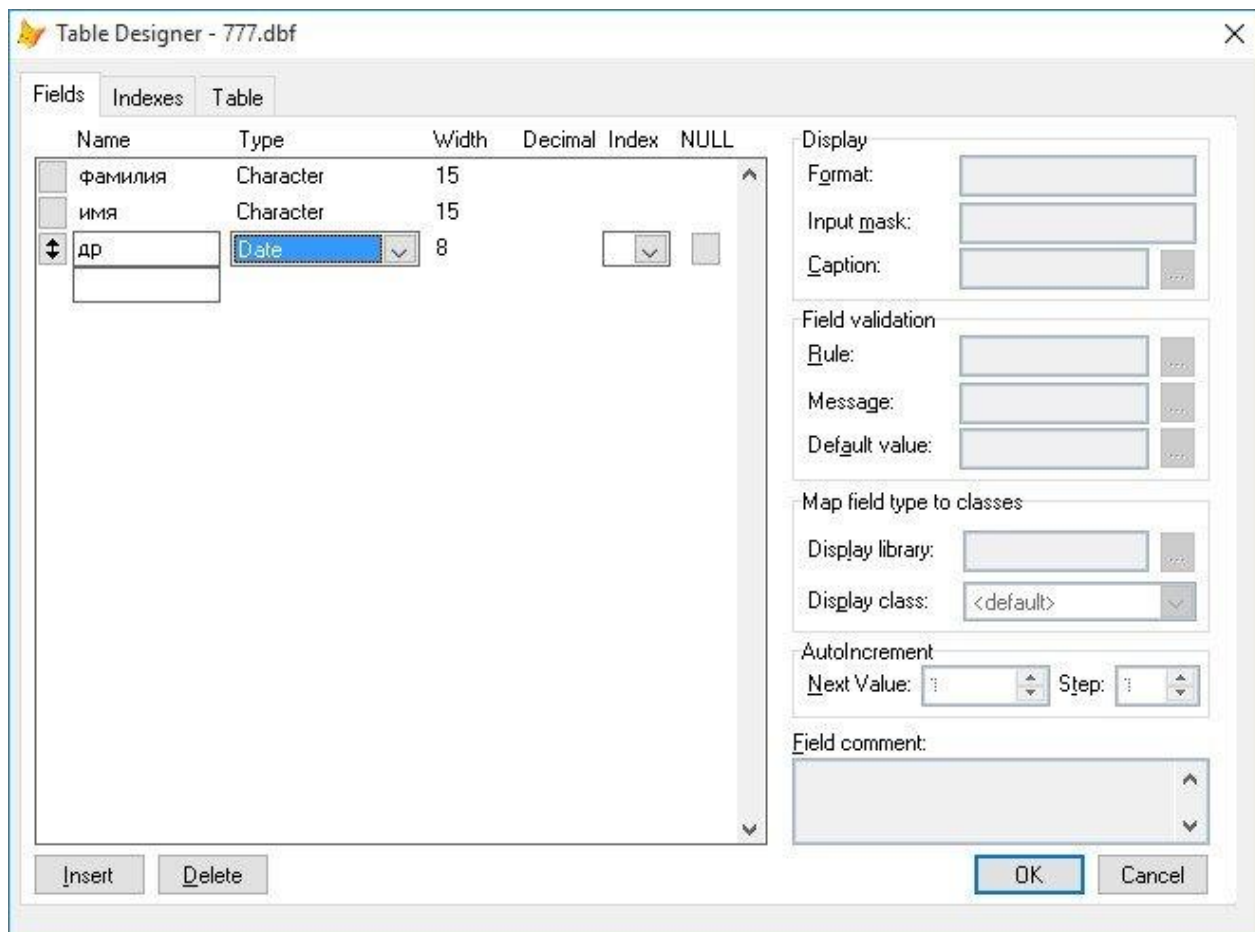


Рисунок 3

Основные типы полей, используемые в Microsoft Visual FoxPro приведены в таблице

Тип поля	Содержимое
Character (Символьный)	Любые символы. Максимальный размер символьного поля – 254 символа
Numeric (Числовой)	Включает цифры от 0 до 9, знак (необязательно), дробную часть (необязательно)
Float (Вещественный)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с обычной точностью
Double (Двойной Вещественный с двойной точностью)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с двойной точностью
Integer (Числовой, целочисленный)	Целые числа
Data (Дата)	Дата, определяющая день, месяц и год. Ввод в это поле контролируется VisualFoxPro автоматически

DataTime (Дата и время)	Дата и время. По умолчанию формат вводимого значения имеет вид мм/дд/гг чч:мм:сс
Logical (Логический)	Логические данные. Может принимать только .Т. – истина, .F. –
	ложь
Мемо (Текстовое поле произвольной длины)	Примечания, размер которых неограничен. Может содержать любую символьную информацию, размер которой ограничивается лишь объемом доступного дискового пространства. Фактически – это указатель на блок данных в файле с расширением frt (файлы dbf и frt одноименны). Для заполнения данного поля нужно дважды щелкнуть мышью по данному полю (или CTRL+Home), после чего в открывшемся окне набрать нужный текст примечаний. Закрыв окно редактирования (CTRL+W), значение поля будет сохранено, а слово мемо в поле базы данных изменится на Мемо.
General (Общие, двоичное поле произвольной длины)	Поле, используется для хранения ссылки на объект. Под объектом понимается любой текстовый файл, звуковой файл, диаграмма, иллюстрация или другой файл, созданный с помощью приложения, поддерживающего технологию OLE. Для заполнения данного поля нужно двойным щелчком мыши на нем открыть окно редактирования. После чего выбрать пункт меню Edit(правка)-insert object(вставить объект) и в открытом выбранном приложении создать нужный объект (например, рисунок). Закрыв окно приложения и окно редактирования поля, поле gen будет заменено на Gen, значение будет сохранено
Currency (Денежный)	Поле для представления денежных сумм, используется для хранения чисел с четырьмя знаками после запятой
Memobinary	Двоичный текст произвольной длины. Содержит любые данные поля мемо, которые можно использовать, не изменяя кодовую страницу
Character binary	Двоичный текст длиной до 254 символов. Содержит любые символьные данные, которые используются, без изменения кодовой страницы

Чтобы визуализировать нашу базу данных надо ввести Browse



Рисунок 4

Далее у нас откроется наша таблица но редактировать мы там нечего не можем. Для этого пишем команду Append blankкаждый раз когда хотим добавить строчку и опять повторяем команду Browse

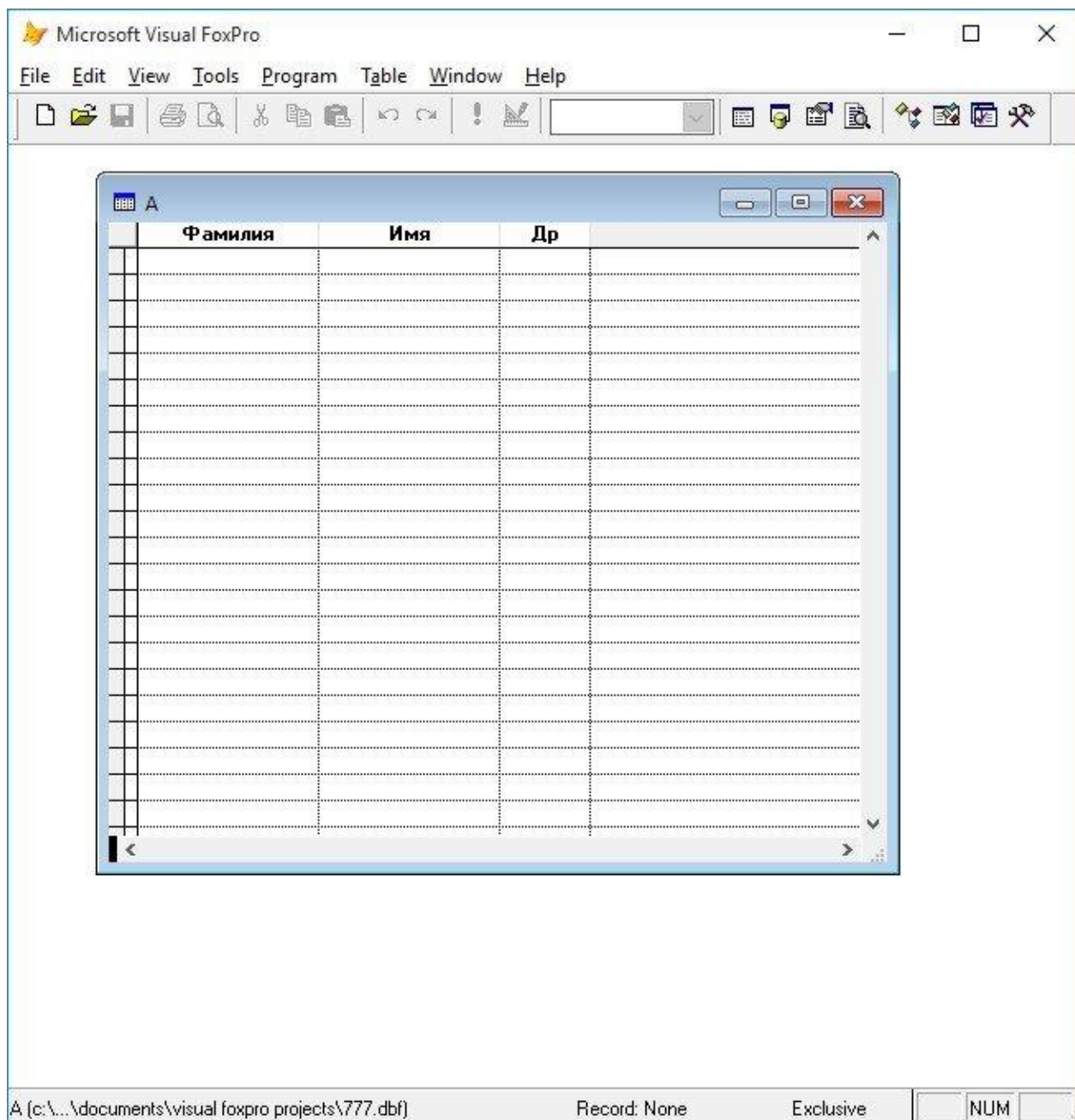


Рисунок 5



Рисунок 6

Всё готово теперь мы можем вносить данных в нашу базу данных. Важно – дата пишется мм/дд/гг

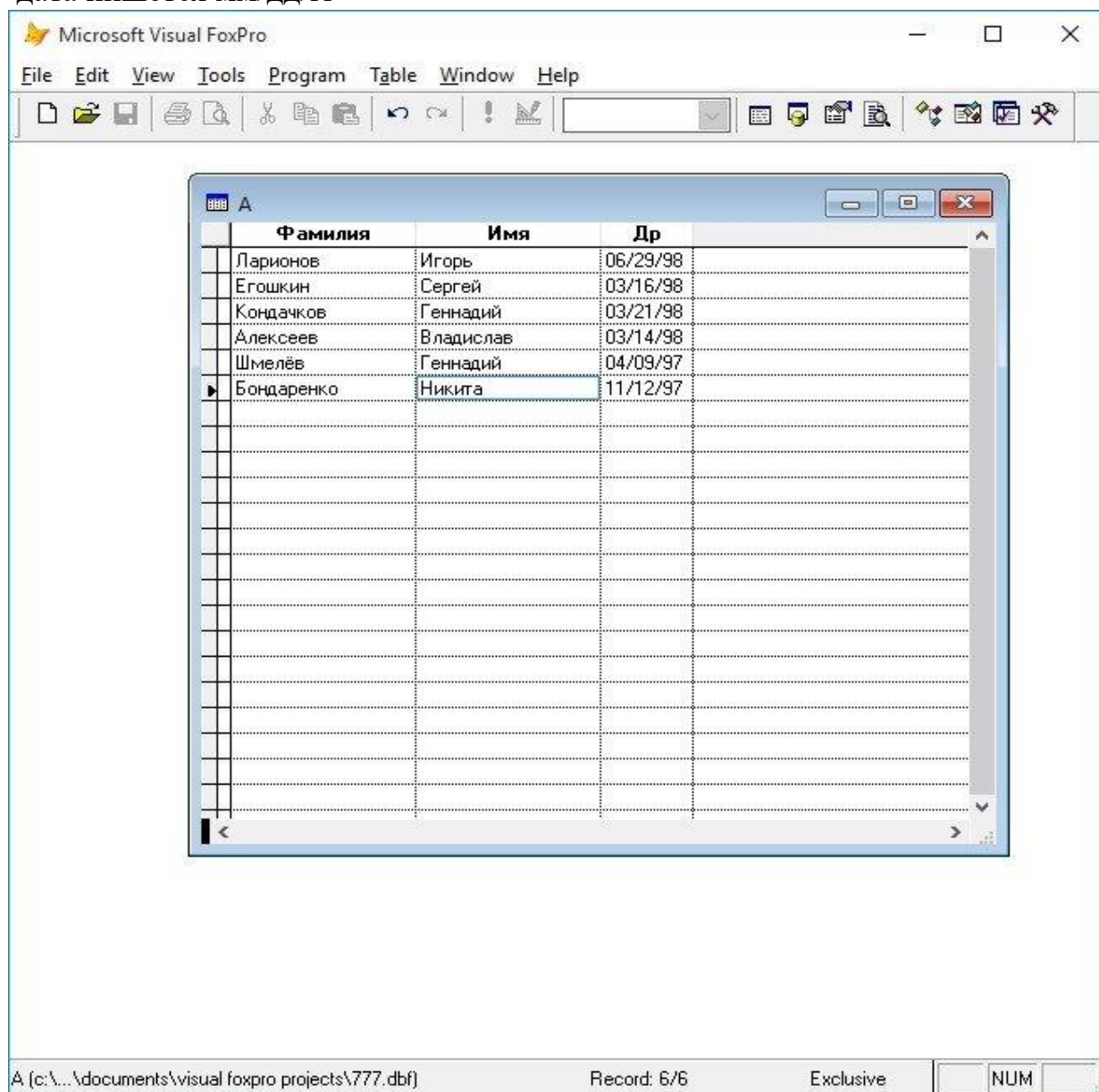


Рисунок 7

Контрольные вопросы:

1. Как создать таблицу в FoxPro?
2. Какие типы полей используются в FoxPro?

Лабораторная работа 6

«Редактирование, добавление и удаление записей в таблице. Применение логических условий к записям. Открытие, редактирование и пополнение табличного файла».

Тема 4 Проектирование структур баз данных

Цель работы: «получить некоторые практические навыки по редактированию базы данных в программе FoxPro»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Команды необходимые для выполнения лабораторной работы

Use<ИМЯ> — открыть подготовить

Select<ИМЯ> — выбрать

Browse — открыть визуализировать

Copy – копирование БД

Rename – переименование БД

Appendblank – добавить запись

Delete record N – пометить на удаление запись N

Pack – удаление помеченных записей

Replace<поле>with<значение> – присвоить значение указанному полю

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Редактирование базы данных

1. Скопировать БД с 5 лабораторной работы;
2. Переименовать БД;
3. Удалить свою запись;
4. Добавить 2-х Ивановых (Петра, Ивана);
5. Добавить поля доход и налог;
6. Заполнить поле доход (произвольно), поле налог 12% от дохода.

Открываем созданную на 5 лабораторной работе нашу БД командой “**Browse**” и выбираем путь, если ваш файл лежит «Не в папке по умолчанию» (Рисунок 1)

Name	Secondname	Birthday	Telephone
Илья	Андреев	04/28/98	9521897603
Владислав	Александров	05/25/98	9512560969
Игорь	Ларионов	12/01/98	9514789986
Геннадий	Шевелев	06/14/98	9521897587
Никита	Бондаренко	04/15/98	9531587987
Владислав	Дубовик	04/01/98	9231234564
Кереев	Евгений	07/03/98	9235490790
Корольев	Алексей	01/01/98	9236580986
Даниил	Никита	02/03/98	9234564967
Кондратов	Геннадий	02/04/98	9231231231
Климентьев	Дмитрий	04/25/98	9216590708
Ястребов	Анастасия	01/25/98	8945867989

Рисунок 1

Задание 1,2

С помощью команды скопируем созданную ранее БД COPY(ИМЯ БД) TO (ПЕРЕИМЕННОВАННОЕ ИМЯ БД) «Писать без скобок» (Рисунок 2)

```

Command
COPY FILE ClarckKent TO ClarckKent2
COPY FILE ClarckKent.dbf TO ClarckKent2.dbf

```

Рисунок 2

(Команда для выполнения копирования вашей БД и автоматическое переименование Новой БД, скопированной) **ОБЯЗАТЕЛЬНО!** Сначала запустите программу БЕЗ ЗАПУСКА ВАШЕЙ БД, иначе будет выдавать ошибку, что «Файл уже используется» После выполнения команды, в папке вы увидите изменения:

Имя	Дата изменения	Тип	Размер
ClarckKent.DBF	09.11.2016 9:50	Microsoft Visual F...	2 KB
clarckkent2.dbf	09.11.2016 9:50	Microsoft Visual F...	2 KB

Рисунок 3 (Результат выполнения команды)

Задание 3

Для того чтобы удалить свою запись нам потребуется команды: *Delete record N* – пометить на удаление запись N, *Pack* – удаление помеченных записей.

Чтобы пометить необходимую вам строку, просто впишите вместо буквы N цифру строки, которую вам необходимо пометить (Рисунок 4)



Рисунок 4 (Результат выполнения команды)

И так, тут мы видим, что 1 строчка в нашей БД помечилась «Черным прямоугольником» Ее нам и надо удалить.

В этом нам поможет команда «Pack-Удаление помеченных записей» (Рисунок 5)

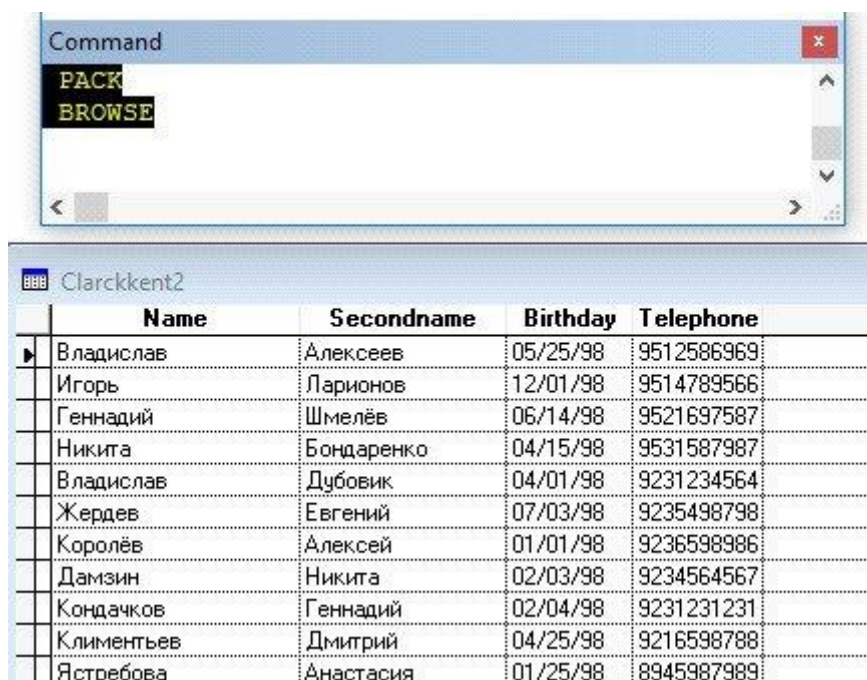


Рисунок 5 (Результат выполнения команды)

Как вы видите, что после выполнения команды Pack, строка пропала и после её выполнения, ваша БД закроется, для этого воспользуйтесь командой “Browse” она откроет вам вашу БД.

Задание 4

Добавим в нашу БД двух Ивановых (Петра и Ивана), с помощью команды *REPLACE name WITH* (Рисунок 6)

Command

```
REPLACE name WITH "Петр"
REPLACE name WITH "Иван"
```

Clarckkent2

	Name	Secondname	Birthday	Telephone
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Петр		/ /	
	Иван		/ /	

Рисунок 6 (Команда для добавления Имени)

Теперь добавим фамилии, при помощи команды *REPLACTE secondname WITH* (Рисунок 7)

Command

```
REPLACE secondname WITH "Иванов"
REPLACE secondname WITH "Иванов"
```

Clarckkent2

	Name	Secondname	Birthday	Telephone
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Петр	Иванов	/ /	
	Иван	Иванов	/ /	

Рисунок 7 (Результат выполнения команды)

Также добавим номер телефона, командой *REPLACE secondname WITH* (Рисунок 8)

Command				
REPLACE telephone WITH 9521697602				
REPLACE telephone WITH 9521645602				

Clarckkent2				
	Name	Secondname	Birthday	Telephone
	Владислав	Алексеев	05/25/98	9521645876
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Петр	Иванов	/ /	9521697602
►	Иван	Иванов	/ /	9521645602

Рисунок 8 (Результат выполнения команды)

Добавляем дату рождения командой *REPLACE birthday WITH* (Рисунок 9)

Command				
REPLACE birthday WITH {^1998-01-02}				
REPLACE birthday WITH {^1998-05-02}				

Clarckkent2				
	Name	Secondname	Birthday	Telephone
	Владислав	Алексеев	05/25/98	9521645876
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	01/02/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Петр	Иванов	01/02/98	9521697602
	Иван	Иванов	05/02/98	9521645602

Рисунок 9 (Результат выполнения команды)

Задание 5

Добавляем поле доход и заполняем его (произвольно) (Рисунок 10)

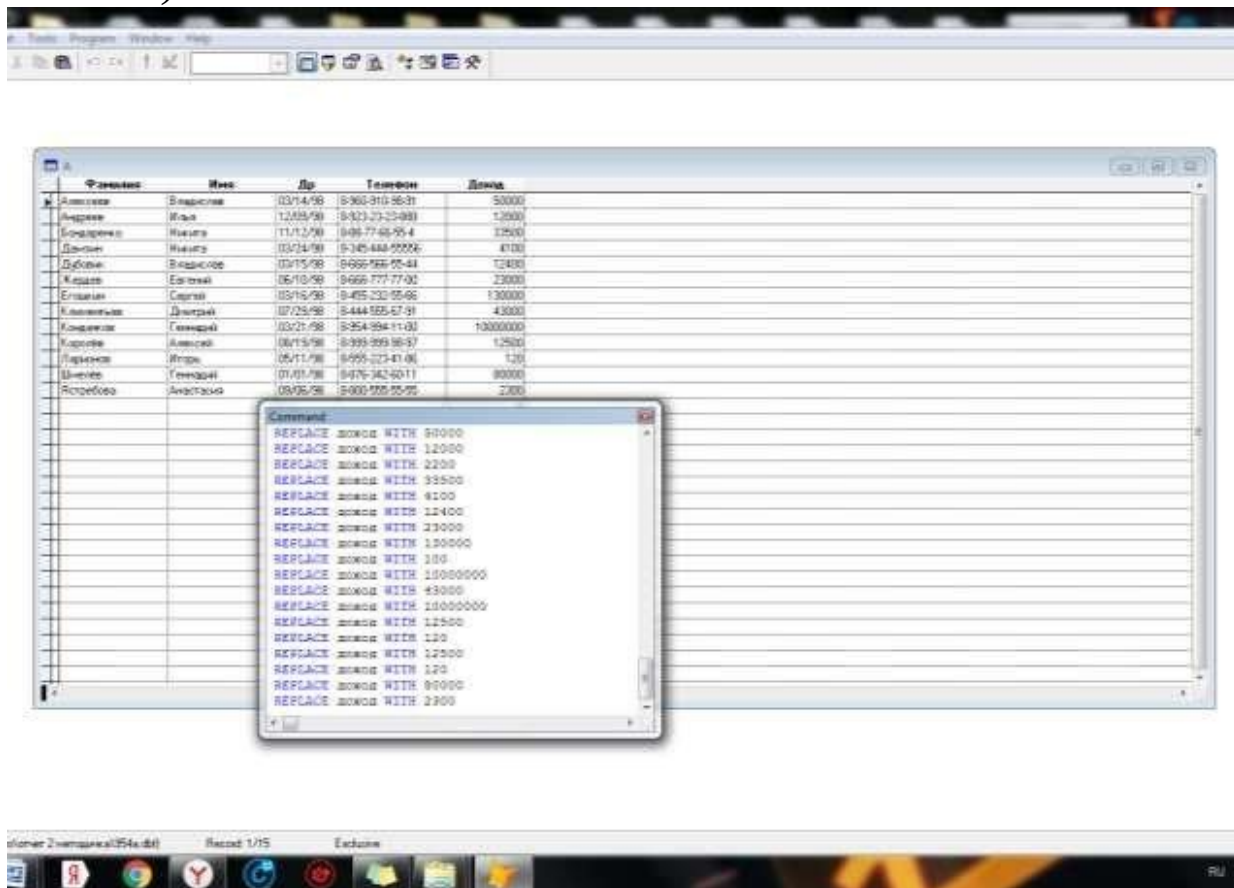


Рисунок 10 (Результат выполнения команды)

Задание 6

Добавляем поле налог, поле налог 12% от дохода (Рисунок 11)

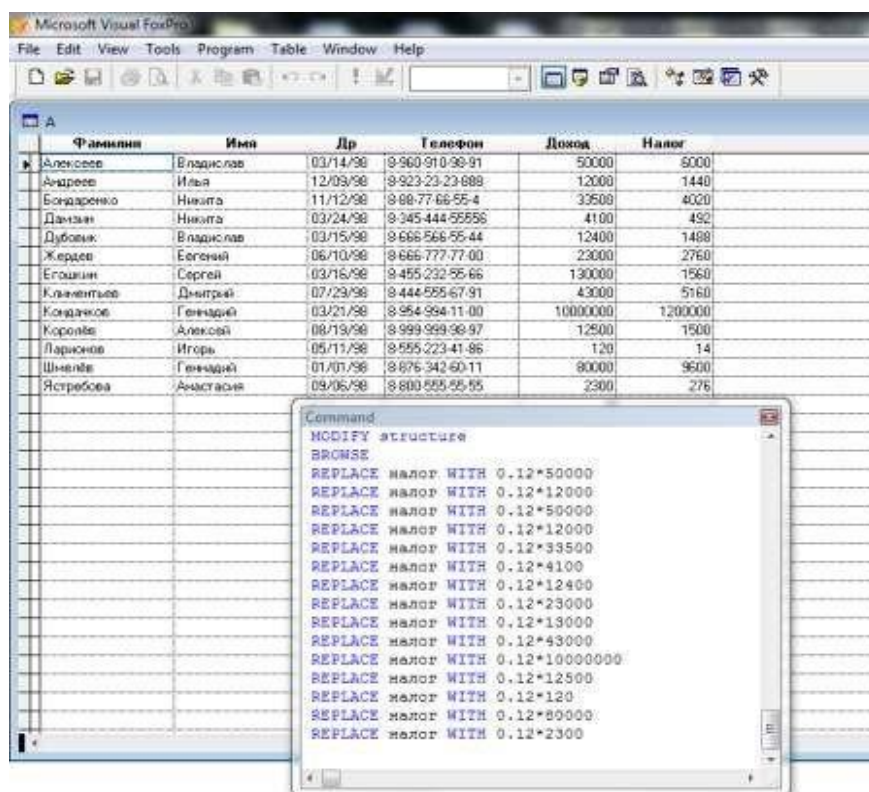


Рисунок 11 (Результат выполнения команды)

Контрольные вопросы:

1. К какому виду СУБД относится FoxPro?
2. С какими командами вы научились сегодня работать? 3. Как добавить новое поле в БД?

Лабораторная работа 7

«Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами».

Тема 4 Проектирование структур баз данных

Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;

- ВЫВОДЫ.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы: Контрольные вопросы:

Лабораторная работа 8

«Создание меню различных видов. Модификация и управление меню».

Тема 4 Проектирование структур баз данных

Цель работы: «получить некоторые практические навыки по созданию меню в программе Visual FoxPro»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения: Команды для создания меню:

Define Menu <имя меню> color r/w – Создание меню

DEFINE PAD <имя пункта> OF <имя меню> PROMPT «название» – Создание горизонтального пункта

ON PAD <имя пункта> OF <имя меню> ACTIVATE POPUP <имя меню раскрывающегося> – Активация горизонтального пункта

DEFINE POPUP <имя меню раскрывающегося> – Создание раскрывающегося пункта

DEFINE BAR 1 OF <имя меню раскрывающегося> PROMPT «название» – Создание вертикального пункта

ON SELECTION BAR 1 OF <имя меню раскрывающегося> do <имя программы> – Активация вертикального пункта

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Создание меню

1. Создать меню;
2. Создать в меню 3 PAD;
3. Создать в каждом PAD POPUP;
4. Создать в каждом POPUP 3 BAR;
5. 1 PAD и 2 PAD, а также в 3 PAD 1 и 2 BAR могут называться как угодно;
6. В 3 PAD 3 BAR должен обязательно называться.
7. Написать программу для BAR «Выход»; 1. Для начала создадим меню.

```
DEFINE MENU Дамзин_Меню COLOR r/w
```

Рисунок – 1 Создание меню.

2. После создания меню нужно добавить горизонтальные пункты.

```
DEFINE PAD p1 OF Дамзин_Меню PROMPT "Вид"  
DEFINE PAD p2 OF Дамзин_Меню PROMPT "файл"  
DEFINE PAD p3 OF Дамзин_Меню PROMPT "Настройки"
```

Рисунок – 2 Создание горизонтальных пунктов.

3. Создадим раскрывающийся пункт.

```
DEFINE POPUP pop1  
DEFINE POPUP pop2  
DEFINE POPUP pop3
```

Рисунок – 3 Создание раскрывающихся пунктов.

4. Теперь нам нужно чтобы при нажатии на горизонтальный пункт появлялся раскрывающийся пункт.

```
ON PAD p1 OF Дамзин_Меню ACTIVATE POPUP pop1  
ON PAD p2 OF Дамзин_Меню ACTIVATE POPUP pop2  
ON PAD p3 OF Дамзин_Меню ACTIVATE POPUP pop3
```

Рисунок – 4 Активация раскрывающихся пунктов.

5. Далее примемся за создание вертикальных пунктов в каждом раскрывающемся пункте.

```
DEFINE BAR 1 OF pop1 prompt "Окно"  
DEFINE BAR 2 OF pop1 prompt "Окно (без рамки) "  
DEFINE BAR 3 OF pop1 prompt "Полный экран"  
DEFINE BAR 1 OF pop2 prompt "Экспорт"  
DEFINE BAR 2 OF pop2 prompt "Импорт"  
DEFINE BAR 3 OF pop2 prompt "Открыть"  
DEFINE BAR 1 OF pop3 prompt "Инструменты"  
DEFINE BAR 2 OF pop3 prompt "Информация"  
DEFINE BAR 3 OF pop3 prompt "Выход"
```

Рисунок – 5 Создание вертикальных пунктов.

6. Напишем программу для закрытия нашего меню.

```

ON SELECTION BAR 3 OF pop3 do exitprog
PROCEDURE exitprog
* activate window wihod
CLOSE ALL
in_stat=.f.
DEACTIVATE POPUP m_menu
quit |

```

Рисунок – 6 Алгоритм закрытия меню.

7. Теперь нам нужно все это сохранить в файл *.prg (По умолчанию сохраняется в C:\Users\Имя пользователя\Documents\Visual FoxPro Projects НА WIN 8.1!!!).

```

SAVE TO ProgramMetoda.prg
DO programmetoda.prg

```

Рисунок – 7 Создание файла *.prg.

8. Через пункт меню file выбираем пункт open и открываем наш файл *.prg и в него уже копируем все из командной строки и сохраняем комбинацией клавиш Ctrl+S.

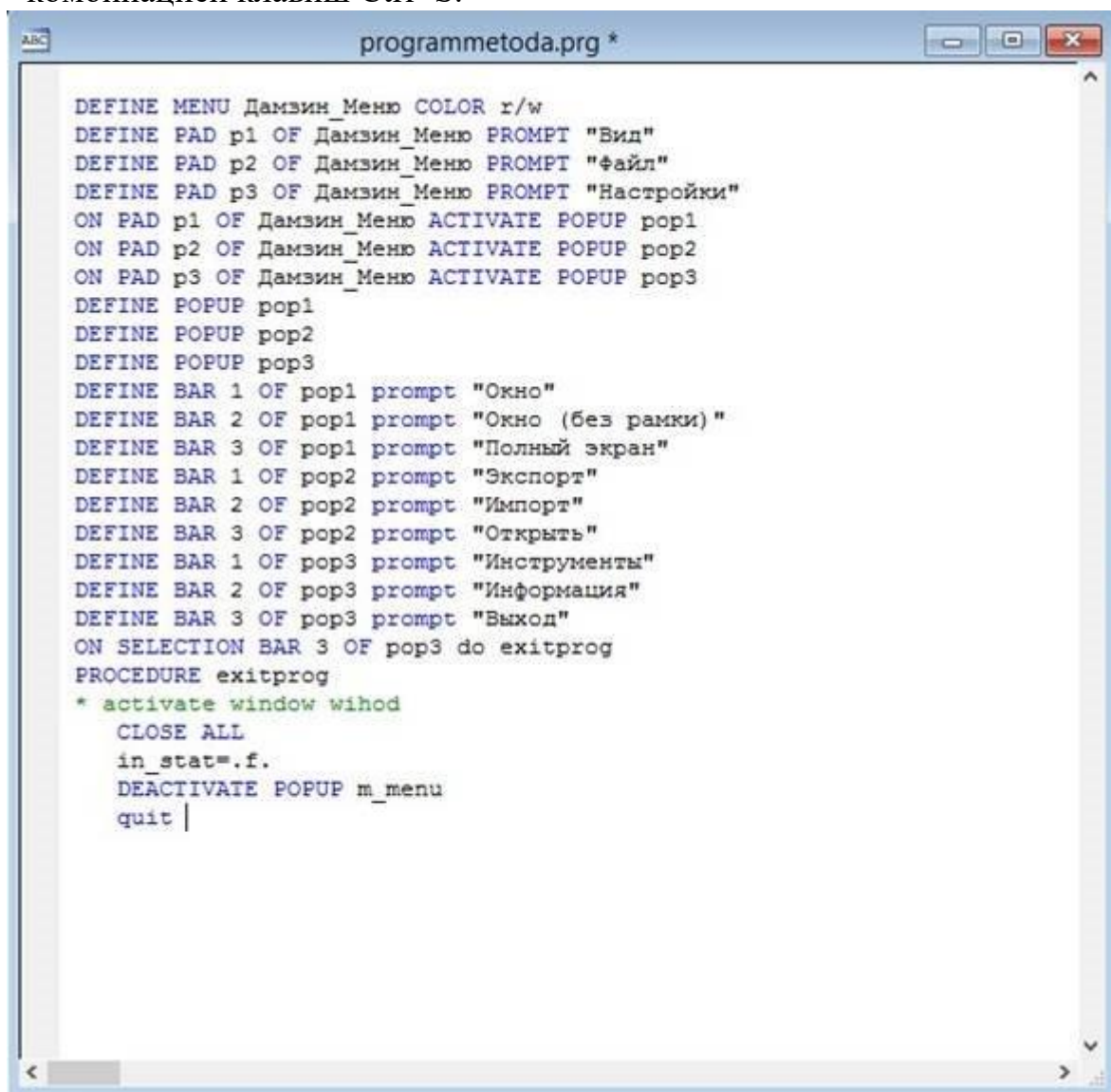


Рисунок – 8 Общий результат работы.

Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE MENU <Название меню>.

Контрольные вопросы:

1. Что такое меню?
2. Перечислите этапы разработки меню в FoxPro?

Лабораторная работа 9

«Создание рабочих и системных окон. Добавление элементов управления рабочим окном»

Тема 4 Проектирование структур баз данных

Цель работы: «Создать в программе меню дополнительные пункты в меню которые будут открывать новые окна с информацией о разработчике и базой данных»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Добавление новых пунктов:

```
DEFINE BAR 6 OF M_MENU PROMPT "Infoprogram"  
DEFINE BAR 7 OF M_MENU PROMPT "openbar"  
  
ON SELECTION BAR 6 OF M_MENU do Infoprogram  
ON SELECTION BAR 7 OF M_MENU do openbar
```

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Создание дополнительных окон в Visual FoxPro

1. Добавляем в наше меню новые пунк-

ты

```
DEFINE BAR 6 OF M_MENU PROMPT "Infoprogram"  
DEFINE BAR 7 OF M_MENU PROMPT "openbar"  
  
ON SELECTION BAR 6 OF M_MENU DO Infoprogram  
ON SELECTION BAR 7 OF M_MENU DO openbar
```

Рисунок 1

2. Добавляем в меню команды выполнения наших программ для новых окон (О создателе и база данных)

```
PROCEDURE infoprogram  
DO program_test.prg  
  
PROCEDURE openbar  
DO oknotest.prg
```

Рисунок 2

3. Пишем код программы для открытия базы данных в новом окне.

```
DEACTIVATE WINDOW all  
RELEASE WINDOWS all  
USE "c:\as\baza.dbf" SHARED  
DEFINE WINDOW f2 FROM 1,1 TO 10,120 COLOR r/b FONT 'arial'  
  
ACTIVATE WINDOW f2  
BROWSE
```

Рисунок 3

4. Пишем код для программы о создателе.

(Если задать шрифт стандартный, то русский текст не будет отображаться корректно)

```
DEFINE WINDOW f3 FROM 12,1 TO 15,35 COLOR r*/w* FONT 'arial'  
  
ACTIVATE WINDOW f3  
@1,1 SAY 'Created by Absolyutly Company'
```

Рисунок 4

Проверяем работоспособность наших пунктов в меню



Рисунок 5

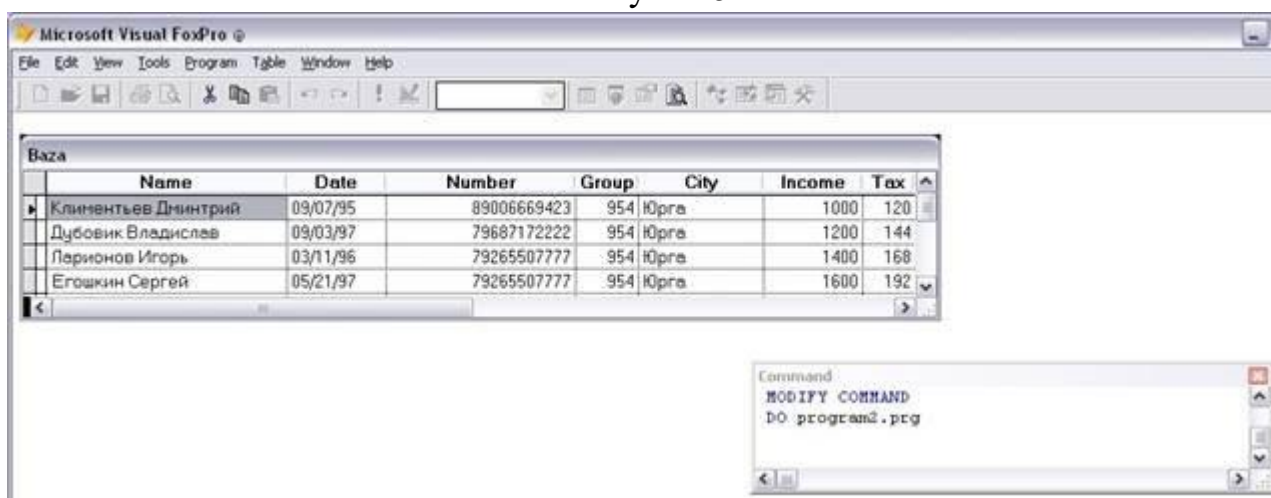


Рисунок 6

Контрольные вопросы:

1. Что такое окно программы?
2. Какие возможности по работе с окном дает FoxPro?

Лабораторная работа 10

«Создание файла проекта базы данных. Создание интерфейса входной формы. Использование исполняемого файла проекта БД, приемы создания и управления».

Тема 4 Проектирование структур баз данных

Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.

2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

1. Создание проекта. Исследование созданного проекта.

Создайте проект любым из имеющихся способов (посредством главного меню или с помощью команды). При сохранении введите имя проекта:

ВНИМАНИЕ! Имя вашего проекта – Myproj_N, где N – ваш номер по журналу (например, если ваш номер 3, то имя проекта - Myproj_3).

Изучите разделы проекта и просмотрите доступные функциональные кнопки проекта.

2. Добавление файлов в проект.

Задание 1.

Перейдите на вкладку **Data** и выберите в списке пункт **Database**. Добавьте в проект базу данных **Mydata**, созданную ранее. Обратите внимание, что при установке курсора на добавленную БД, в нижней части проекта отображается полный путь к файлу БД. ***Убедитесь, что путь соответствует вашей папке!***

Проверьте работу функциональных кнопок на данной вкладке.

На вкладке **Data** выберите пункт **Free Tables** (свободные таблицы) и добавьте в этот раздел свободную таблицу вашего варианта, созданную на первом практическом занятии. ***Убедитесь, что путь таблицы соответствует вашей папке!*** Проверьте работу функциональных кнопок при работе со свободными таблицами.

Скопируйте скрин-шот с вкладкой Data в файл отчета.

Задание 2.

Перейдите на вкладку **Documents**. Изучите активные функциональные кнопки.

Задание 3.

Перейдите на вкладку **Code**. Изучите активные функциональные кнопки. Выберите в списке пункт **Program**. Добавьте все разработанные вами программы в проект.

Убедитесь, что пути программных файлов соответствуют вашей папке! Проверьте работу функциональных кнопок при работе с программными файлами. Используя кнопки, откройте любую программу для изменения, попробуйте запустить программу и т.д.

Используя кнопки проекта, создайте новую программу **prog20**, которая решает задачу, выданного вам варианта (на карточке).

Текст созданной программы и результат ее выполнения добавьте в отчет. Проверьте наличие созданной программы в проекте. Если программа отсутствует, то самостоятельно добавьте ее в проект. Скопируйте скриншот с вкладкой Code в файл отчета.

Задание 4.

Перейдите на вкладку **Other**. Изучите активные функциональные кнопки. Выберите в списке пункт **Other Files**. Добавьте в проект несколько картинок (файлы с расширением .bmp, .jpg и т.д.).

Скопируйте скриншот с вкладкой Other в файл отчета.

3. Покажите результат преподавателю. Закройте созданный проект. После выхода в операционную систему проверьте наличие в вашей папке файлов с

Для возможности проверки преподавателем ваших выполненных заданий, необходимо создать файл отчета в среде Microsoft Word, в который скопировать все выполненные команды и скрин-шоты с результатами.

вашим именем Myproj_N.pjx и Myproj_N.pjt.

«Создание запросов»

Откройте свой проект, созданный ранее.

1. Создание запроса с помощью мастера из главного меню.

Задание: создать запрос на основе таблицы **knig**, содержащий поля **Название** и **Автор**.

В главном меню выберите пункт **File – New** и в окне **New** выберите создание запроса с помощью мастера. В окне **Wizard Selection** выберите мастер запросов:



Выполните шаги мастера:

1 шаг – из таблицы **Knig** выберите поля **nazvanie** и **autor**.

2 шаг – здесь отсутствует, так как запрос создается на основе одной таблицы.

3 шаг – оставьте без изменения.

4 шаг – установите сортировку по полю *nazvanie* по возрастанию.

5 шаг – завершите создание запроса.

Сохраните запрос под именем **knig1**. Проверьте наличие созданного запроса в своей папке.

2. Запуск запроса на выполнение разными способами.

В командном окне выполните команду **DO knig1.qpr** и посмотрите созданный запрос.

Добавьте в свой отчет скрин-шот с командой и результатом ее выполнения.

Затем попробуйте запустить запрос, используя опции главного меню – **Program – Do**.

Добавьте созданный запрос в свой проект. Для этого перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **Add**, выберите имя созданного вами запроса и нажмите на кнопку **Ok**.

Запустите запрос из проекта – кнопка **Run**.

3. Создание запроса с помощью команды **CREATE QUERY** в режиме конструктора.

*Задание: создать запрос на основе таблицы **knig**, содержащий поля **Название**, **Автор** и **Цена** и записи с ценой больше 100.*

В командном окне выполните команду **CREATE QUERY**. В результате запускается конструктор запросов.

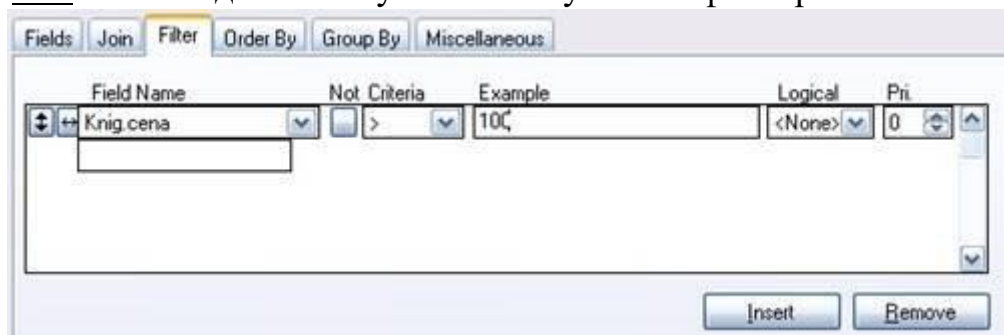
1 шаг. Выберите таблицу, на основе которой будет создан запрос – **knig**, нажмите кнопку **Add** и закройте окно **Add Table or View**.

Для удобства работы в конструкторе запросов используется панель инструментов



2 шаг. На вкладке **Fields** выберите поля, включаемые в запрос (см. задание). Для этого выберите нужное поле и нажмите кнопку **Add**.

3 шаг. На вкладке **Filter** установите условие фильтра:



Для создания используйте разворачивающиеся списки.

4 шаг. На вкладке **Order By** выберите условие сортировки – по полям **autor** и **pazvanie**.

5 шаг. Для просмотра запроса нажмите кнопку  на панели инструментов. *Добавьте в свой отчет скрин-шот с результатом выполнения запроса.*

6 шаг. С помощью кнопки  панели инструментов просмотрите текст программы, которая создает запрос. *Добавьте в свой отчет скрин-шот с программным кодом.*

7 шаг. Закройте окно конструктора запросов. Сохраните запрос под именем **knig_cen**. Добавьте созданный запрос в свой проект.

Добавьте созданный запрос в свой проект.

Запустите созданный запрос любым способом и просмотрите результат.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

4. Изменение существующего запроса.

Выберите в проекте имя созданного вами запроса **knig_cen** - и нажмите на кнопку **Modify**. В результате открывается конструктор запросов.

На вкладке **Filter** измените условие отбора записей: **cena<100**. Сохраните запрос и просмотрите результат.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

5. Создание запроса из проекта с помощью конструктора.

Перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **New**. В появившемся окне нажмите кнопку **New Query**.

Далее аналогично предыдущему запросу создайте запрос: на основе таблицы **Tip**. В запрос выберите поле **name** и установите сортировку по этому же полю. Сохраните запрос с именем **tip1** и просмотрите его.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

6. Создание запроса на основе двух таблиц с помощью мастера.

Задание: создать запрос на основе двух таблиц - **tip** и **knig**, содержащий поля

Наименование типа, Название книги, Автор и Цена.

Перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **New**. В появившемся окне нажмите кнопку **Query Wizard**. В результате будет запущен мастер создания запросов.

1 шаг – Выбор полей. Из таблицы **Tip** выберите поле **name**, из таблицы **Knig** выберите поля **Autor**, **Nazvanie**, **Cena**.

2 шаг – Установление связи между таблицами. Нажмите кнопку **Add** для выбора связи.

2а шаг – Записи, включаемые в запрос. Оставляем без изменения.

3 шаг – Фильтрация записей. Для эксперимента установите фильтр для таблицы

Tip

и нажмите кнопку только записи, у которых **kodt** равен 1. Очистите условие фильтра и перейдите на следующий шаг.

4 шаг – Сортировка записей. Установите сортировку по полю **name**.

4а шаг – Лимит записей. Оставляем без изменения.

5 шаг – Финиш. Нажмите кнопку **Finish** и сохраните запрос с именем **tip_kn**.

Запустите запрос для просмотра. Обратите внимание, что в нем содержатся данные из двух таблиц, что очень удобно для просмотра.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

Нажмите в проекте кнопку **Modify** для созданного запроса и перейдите на вкладку **Join**. Проверьте, правильно ли установлена связь между таблицами.

7. Покажите результат преподавателю. Закройте свой проект и осуществите выход из системы.

Контрольные вопросы:

1. Что такое проект в FoxPro? 2. Какие виды проектов существуют? 3. Каким способом можно создать проект в БД?

Лабораторная работа 11

«Создание формы. Управление внешним видом формы».

Тема 4 Проектирование структур баз данных

Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

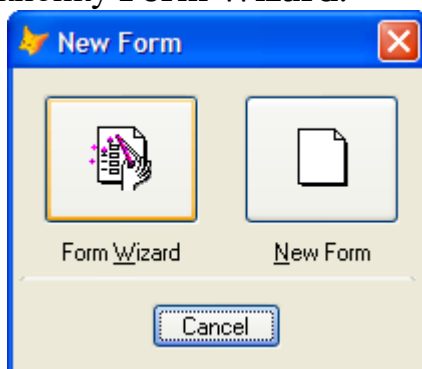
Задания для выполнения лабораторной работы:

«Создание форм с помощью мастера» *Откройте свой проект, созданный ранее.*

1. Создание однотабличной формы.

Задание: создать с помощью мастера форму на основе таблицы **izd**.

Перейдите в проекте на вкладку **Documents** и выберите раздел **Forms**. Для создания новой формы нажмите кнопку **New** в проекте. В появившемся окне **New Form** нажмите кнопку **Form Wizard**:

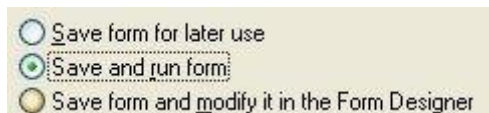


В появившемся окне **Wizard Selection** выберите пункт **Form Wizard** для создания однотабличной формы:



Выполните шаги мастера, выполняя все его инструкции.

- 1 шаг – для таблицы **Izd** выберите все поля.
- 2 шаг – в качестве оформления формы выберите стиль **Shadowed** и тип кнопок **Custom**.
- 3 шаг – установите сортировку по полю **n_izd** по возрастанию.
- 4 шаг – введите заголовок формы – **Данные об издательствах**, просмотрите результат с помощью кнопки **Preview**. Перед сохранением выберите в окне опцию **Save and run form** (Сохранить и запустить форму).



После этого нажмите кнопку **Finish**.

В появившемся окне **Save As** укажите путь к своей папке и задайте имя новой формы – **izd1**.

Если все действия были выполнены верно, то форма будет сохранена и запущена. Проверьте как осуществляется вывод данных в форме, используя полосу прокрутки. Проверьте работу кнопок.

Добавьте в таблицу новую запись с данными: код издательства – 1, название издательства – **Форум**, адрес – **Москва**, фио руководителя – **Кузьмичев А.С.**

2. Модификация формы.

Задание: внести изменения в форму **izd1**, используя конструктор.

Откройте созданную форму **izd1** в режиме конструктора.

- *изменение цвета объектов формы:*

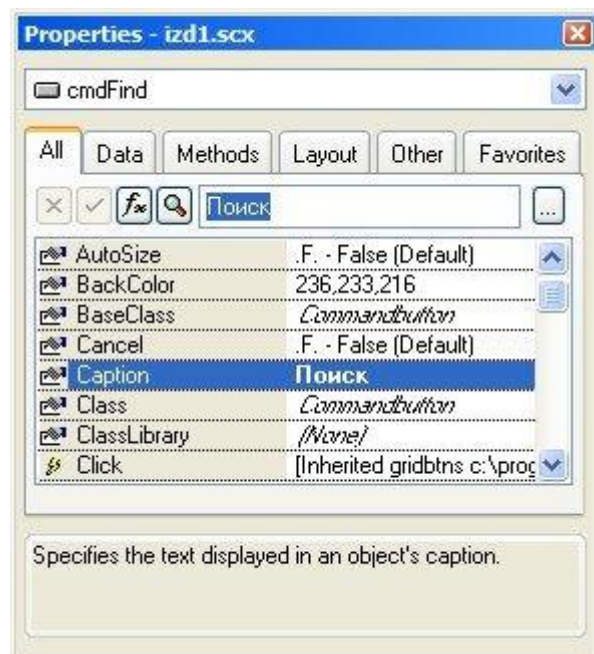
Активируйте панель инструментов **Color Palette** с помощью пункта меню **View – Toolbars**. С помощью кнопок данной панели измените цвет надписи «Данные об издательствах» и цвет фона формы. Запустите форму, сохраните ее и проверьте результат внесенных изменений. - *изменение расположения и размеров объектов формы:*

Переместите надпись «Данные об издательствах» так, чтобы она располагалась по горизонтали по центру.

Уменьшите размер контейнера, в котором отображаются данные. Запустите форму и проверьте результат внесенных изменений.

- *изменение свойств объектов:*

В окне **Properties** выберите в списке объектов формы объект **cmdFind**. Затем в списке свойств этого объекта найдите свойство **Caption** и введите в строке ввода слово **Поиск**.



В результате в форме эта кнопка будет иметь название «Поиск». Аналогично измените названия остальных кнопок: **cmdPrint** - Печать, **cmdAdd** - Ввод, **cmdDelete** - Удаление, **cmdExit** – Выход.

Для кнопки Выход установите свойство **FontBold** равным **.T.**, свойство **ForeColor** равным **255,0,0**. Обратите внимание, что значения всех свойств, которые мы изменяли в окне **Properties**, изменяют свой цвет на **фиолетовый**. Запустите форму и проверьте результат внесенных изменений. Закройте форму и сохраните все изменения.

3. Создание формы на основе двух таблиц.

Задание: создать с помощью мастера форму на основе таблиц **tip** и **knig**.

Запустите мастер форм из проекта. В окне **Wizard Selection** выберите пункт **Oneto-Many Form Wizard** для создания формы один-ко-многим:



Выполните шаги мастера:

- 1 шаг – для родительской таблицы **tip** выберите поле **naim**.

2 шаг – для дочерней таблицы **knig** выберите поля **autor, nazvanie, cena, kol**. 3 шаг – проверьте, правильно ли установлена связь между таблицами – должна быть установлена связь по полю **kodt**.

4 шаг – в качестве оформления формы выберите стиль **Embossed** и тип кнопок **Picture buttons**. 5 шаг – установите сортировку по полю **Name**.

6 шаг - введите заголовок формы – **Перечень книг**, просмотрите результат с помощью кнопки **Preview**.

После этого нажмите кнопку **Finish**.

В появившемся окне **Save As** укажите путь к своей папке и задайте имя новой формы – **tip_knig**.

4. Тестирование формы.

Задание: запустить форму **tip_knig** и проверить ее работу.

Запустите форму из проекта. Проверьте работу всех кнопок. Обратите внимание на то, какие недочеты имеет форма, чтобы в дальнейшем их устранить.

5. Модификация формы.

Задание: внести изменения в форму **tip_knig**, используя конструктор.

Откройте созданную форму **tip_knig** в режиме конструктора. Откройте все необходимые для работы панели инструментов.

- изменение размеров объектов:

Так как данные таблицы не полностью видны в форме, то необходимо изменить размер формы (увеличить по ширине) и размер объекта **Grid1** так чтобы отображались все данные. С помощью мыши измените размеры указанных объектов.

Запустите форму, сохраните ее и проверьте результат внесенных изменений.

- изменение свойств объектов:

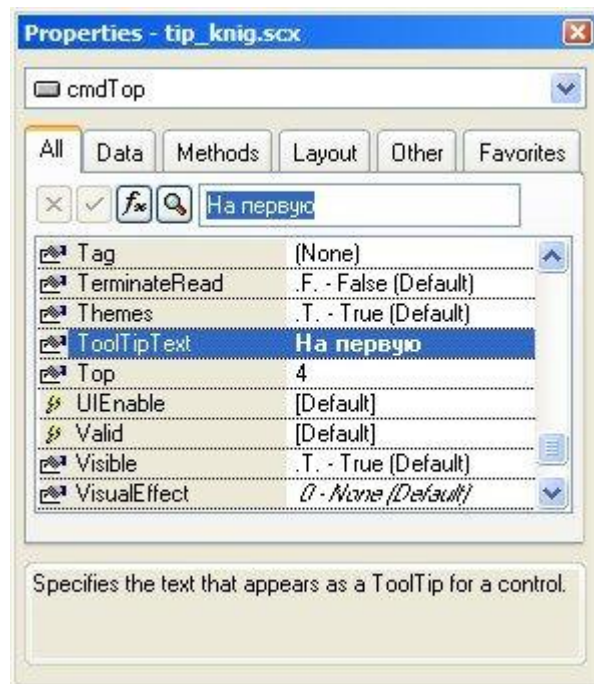
В окне **Properties** выберите в списке объектов формы объект **Form1**. Затем в списке свойств этого объекта найдите свойство **MaxButton** и измените его значение на **.F.** . Аналогично для свойства **MinButton** установите значение **.F.** . Обратите внимание, что в заголовке формы исчезли кнопки максимизации и минимизации окна.

Запустите форму, сохраните ее и проверьте результат внесенных изменений.

Для надписи **Перечень книг** измените цвет текста и расположите ее по горизонтали по центру. Для текстового поля **Name** и подписи к нему измените размер шрифта и цвет текста.

При запуске формы обратите внимание, что для каждой кнопки в форме имеется всплывающая подсказка, но на английском языке. Это не очень удобно.

В окне **Properties** выберите в списке объектов формы объект **cmdTop**. Затем в списке свойств этого объекта найдите свойство **ToolTipText** и введите в строке ввода слово **На первую**.



Запустите форму и проверьте измененную подсказку. Аналогично измените текст подсказки для всех остальных кнопок – **Предыдущая, Следующая, На последнюю, Поиск, Печать, Ввод/Сохранение, Редактирование, Удаление, Выход.**

Запустите форму, сохраните ее и проверьте результат внесенных изменений. Завершите работу с формой **tip_knig**.

6. Покажите результат работы преподавателю.

«Конструктор форм. Настройка параметров формы. Определение среды окружения»

1. Создание экранной формы с помощью команды *CREATE FORM*. Задание: создать форму, настроить основные свойства формы.

В командном окне выполните команду **CREATE FORM f1**. В результате запускается конструктор форм и открывается окно со свойствами формы.

Для работы с конструктором форм необходимо окно свойств **Properties**, панели инструментов **Form Designer** и **Form Controls**.

Панель Form Designer

Панель инструментов Form Designer автоматически отображается вместе с конструктором форм.

Окно свойств **Окно Properties** открывается со списком свойств и событий выделенного объекта и используется для установления его свойств.

Панель **Form Controls** С помощью панели инструментов Form Controls можно создавать элементы управления в формах

Если окно свойств отсутствует, то нажмите

кнопку  на главной панели инструментов.

Если панели инструментов отсутствуют, то для их активации выберите пункт меню **View – Toolbars** и выберите нужную панель.

В окне свойств установите основные свойства формы:

AutoCenter – .t.	центрирование объекта Form
BorderStyle – 1	стиль обрамления объекта
Caption – «Типы книг»	текст названия объекта
MaxButton – .f.	Кнопка максимизации объекта
MinButton – .f.	Кнопка минимизации объекта
Picture	Установка фонового рисунка
WindowType – 0	0 – немодальный, меню активно. 1 – модальный. В этом случае будет активна только данная форма, никакой другой объект Form не может стать активным, меню неактивно.

Запустите форму для проверки установленных свойств.

Для запуска формы используется кнопка  на главной панели инструментов.

2. Определение среды окружения формы и вставка в форму элемента «Сетка».

Задание: подключить к форме таблицу *tip*, добавить в форму элемент *Grid*, настроить необходимые свойства.

Для подключения данных к форме используется диалоговое окно **Data Environment**. Оно отображает элементы среды данных, с которой работает экранная форма. Для активации окна **Data Environment** используются следующие способы: через главное меню **View**, через контекстное меню, через панель инструментов **Form Designer** – кнопка .

Используйте любой из предложенных способов – на экране появится окно для выбора нужной таблицы. Выберите таблицу **tip** и закройте окно. Выбранная таблица будет вставлена в окно **Data Environment**.

Выделите таблицу и установите для нее свойство **Order** – индекс **t1**. Данное свойство подключает индекс для сортировки таблицы.

Для вставки таблицы в форму достаточно перетащить таблицу за заголовок мышкой на форму. Выполните это. В результате в форме создается элемент управления **Grid**. После этого закройте окно **Data Environment**, так как оно нам пока не потребуется.

Рассмотрите в окне свойств **Properties** свойства элемента **Grid**, которые установлены **по умолчанию** (они выделены фиолетовым цветом).

RecordSource	Задаёт источник данных, к которому привязан элемент управления
--------------	--

	Grid.
--	-------

Установите дополнительные свойства:

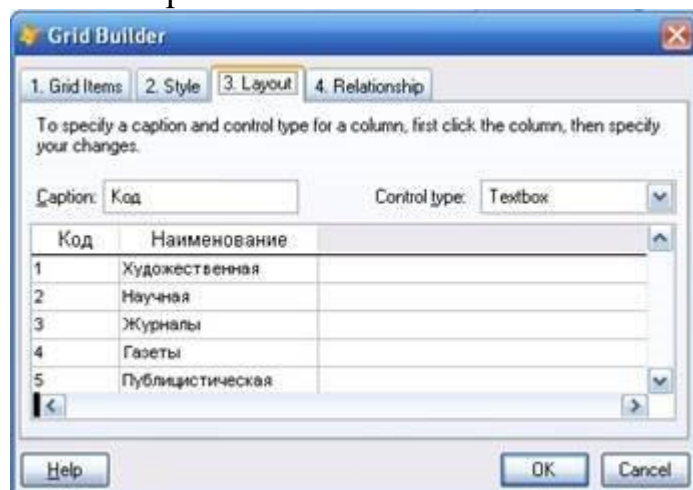
DeleteMark - .f.	Определяет, будет ли в элементе управления Grid отображаться столбец маркеров удаления.
ReadOnly - .t.	Определяет, может ли пользователь редактировать элемент управления или разрешены ли обновления таблицы или представления, связанных с объектом Cursor. .t. - Пользователь не может редактировать элемент управления.

Для настройки элемента **Grid** можно щелкнуть по этому элементу правой кнопкой мыши и использовать команду **Builder...** (построитель) из контекстного меню. Выберите эту команду.

На вкладке **Grid Items** указаны выбранные поля.

На вкладке **Style** выберите по желанию стиль оформления элемента **Grid**.

На вкладке **Layout** в разделе Caption введите заголовки столбцов таблицы: 1 столбец – **Код**, 2 столбец – **Наименование**. Здесь же можно изменить ширину столбцов. 1 столбец сделайте поуже, а 2 столбец – пошире, чтобы данные не обрезались.



На вкладке **Relationship** устанавливаются связи между таблицами. Так как у нас в форме будет отображаться одна таблица, то здесь ничего не делаем.

После установленных изменений нажмите кнопку **Ok**.

Измените размеры элемента **Grid** – уменьшите ширину так, чтобы были видны только столбцы таблицы. Разместите таблицу в центре формы.

Запустите форму для просмотра и проверьте внесенные изменения.

3. Вставка в форму элемента «Кнопка».

Задание: создать в форме кнопку «Выход», при нажатии которой форма будет закрываться.

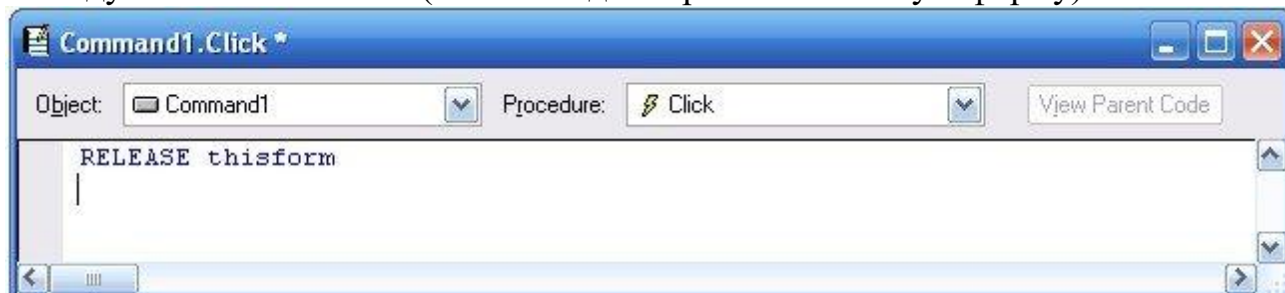
Чтобы вставить элемент Кнопка, щелкните на изображение кнопки  на панели **Form Controls** (Command Button), затем установите указатель мыши в форме и щелкните для вставки элемента управления. В форме появится элемент кнопка с именем **Command1**.

Для элемента Кнопка установите следующие свойства:

Caption – «Выход»	Название кнопки
FontSize – 12	Задаёт размер шрифта для текста, отображаемого вместе с объектом
ForeColor – выбрать красный цвет	Задаёт основной цвет символов, используемые при отображении текста и графики в объекте.
FontBold – .t.	Полужирное начертание букв
Cancel – .t.	Определяет, является ли элемент управления CommandButton кнопкой Cancel; это означает, что если пользователь нажмет клавишу ESC, произойдет событие Click для кнопки Cancel

Для того, чтобы кнопка работала в форме, необходимо обработать событие, которое будет происходить при нажатии кнопки. В кнопках всегда обрабатывается событие **Click** – однократное нажатие левой кнопки мыши.

Щелкните два раза по кнопке **Command1**. Откроется окно **Command1.Click** для ввода обрабатывающей процедуры. Введите одну команду - **release thisform** (эта команда закрывает активную форму).



Закройте окно **Command1.Click**.

Запустите форму и проверьте работу созданной кнопки.

4. Настройка порядка обхода элементов управления в форме.

Задание: проверить и настроить порядок обхода элементов в форме.

Для настройки порядка обхода элементов управления в форме используется кнопка **Set Tab Order**  на панели инструментов **Form Designer**.

2. Нажмите эту кнопку и установите порядок элементов: сетка – 1, кнопка –



Нажмите кнопку повторно. Порядок обхода сохранится. Запустите форму для просмотра.

Как видите, при запуске формы сразу активна таблица и по ней можно перемещаться с помощью стрелок вверх и вниз. Для выхода из формы достаточно нажать кнопку **Выход**.

5. Добавление в форму второй кнопки.

Задание: добавить в созданную форму кнопку «Ввод новой записи», при нажатии на которую будет загружаться новая форма – для ввода новой записи в таблицу.

Добавьте в форму новую кнопку - **Command2**. Установите нужные свойства:

Caption – «Ввод новой записи».

Двойным щелчком мыши вызовите окно **Command2.Click**. Введите следующие команды:

```
do FORM f2 && запуск новой формы
thisform.grdTip.SetFocus    && установка фокуса на элемент Сетка
thisform.Refresh           && обновление формы
```

Здесь после знака && введены комментарии. Закройте окно **Command2.Click**.

Созданная кнопка пока не будет работать, так как у нас не создана форма **f2**. Если попробовать запустить форму и нажать данную кнопку то выйдет ошибка.

Сохраните форму. Закройте конструктор форм.

Созданную форму необходимо добавить в проект. Перейдите в проекте на вкладку **Documents** и выберите пункт **Forms**. Нажмите кнопку **Add** и добавьте созданную форму в свой проект.

6. Создание экранной формы из проекта.

Задание: создать форму, установить основные свойства, подключить данные, создать коды обработки событий.

В проекте перейдите на вкладку **Documents** и выберите пункт **Forms**.
Нажмите кнопку **New - New Form**.

Задайте основные свойства формы: AutoCenter – .t.

BorderStyle – 1

Caption – название формы «Ввод новой записи» MaxButton – .f.

MinButton – .f.

WindowType – 1

Подключите к форме таблицу **tip** с помощью окна **Data Environment**.
Выделите таблицу и установите для нее свойство **Order** – индекс **t1**. Перетащите в форму из окна **Data Environment** из таблицы **tip** поле **name**. В результате в форму будут вставлены два элемента – **Label** (текст) и **Text Box** (поле для ввода).

Необходимо отредактировать данные элементы.

Выделите элемент **lblName** и измените для него свойство **Caption** – Наименование.

Передвиньте элемент так, чтобы он не закрывал поле для ввода.

Выделите элемент **txtName** и измените для него свойство **ControlSource** – m.name.

Это свойство задает источник данных, к которому привязывается объект.
В нашем случае осуществляется привязка к переменной памяти, которая будет создана командой SCATTER MEMVAR BLANK.

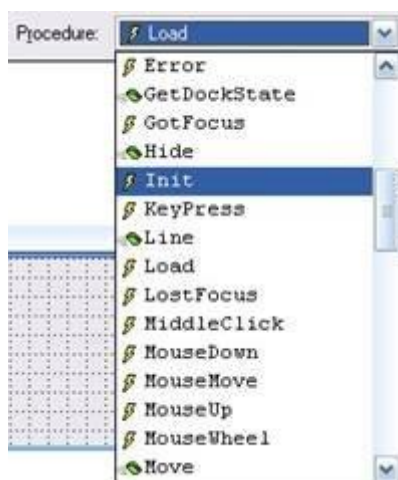
Справка:

Команда SCATTER MEMVAR - копирует содержимое текущей записи таблицы в набор переменных памяти. При ссылке на переменную памяти, чтобы отличать ее имя от имени соответствующего поля в таблице, ставьте перед именем переменной префикс 'm.'.

Команда SCATTER MEMVAR BLANK - Создает набор пустых переменных памяти.

Для данной формы необходимо создать код для обработки события загрузки формы – это событие **Init**. Чтобы мы могли ввести новую запись в таблицу **tip**, необходимо создать пустой набор переменных памяти при загрузке формы.

Выполните двойной щелчок мыши по любому свободному месту формы. Откроется окно для обработки кодов формы – **Form1.Load**. Выберите в разделе **Procedure** из разворачивающегося списка событие **Init**:



Обратите внимание, что у окна поменяется название на **Form1. Init**. Введите команду **SCATTER MEMVAR blank** и закройте окно. **Сохраните форму с именем f2.**

7. Вставка в форму кнопок.

Задание: добавить в форму кнопки «Выход» и «Запись».

Добавьте в форму кнопку «**Выход**» аналогично первой форме.

Добавьте в форму кнопку «**Запись**», при нажатии которой будет осуществляться запись новых данных в таблицу. Поэтому для данной кнопки нужно ввести код для события **Click**:

APPEND BLANK

REPLACE naim WITH thisform.TxtName.Value

RELEASE thisform

Настройте порядок обхода элементов управления в данной форме: элемент **txtName** – 1, кнопка **Запись** – 2, кнопка **выход** - 3.

Сохраните форму.

Для проверки работы созданных форм запустите форму **f1** (кнопка Run проекта), в ней нажмите кнопку «Ввод новой записи». В появившемся окне введите произвольную новую запись.

8. Покажите результат работы преподавателю.

9. Ответьте на контрольные вопросы.

Контрольные вопросы:

1. Для создания формы используется команда ...
2. Окно Properties используется для ...
3. Для создания среды окружения формы используется...
4. Панель инструментов Form Controls используется для ...
5. Событие Click означает - ...
6. Событие Init означает - ...
7. Элемент Grid используется для ...
8. Элемент Label используется для ...
9. Элемент TextBox используется для ...

Лабораторная работа 12

«Задание значений и ограничений поля. Проверка введенного в поле значения. Отображение данных числового типа и типа дата»

Тема 4 Проектирование структур баз данных

Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

«Использование стандартных функций Visual Foxpro в программировании»

1. Использование математических функций.

Создайте программу prog16. В программе выполните следующие действия: Переменной *x* присвойте произвольное вещественное значение с числом десятичных знаков, равное 5 (например, 123.67953).

Используя встроенные математические функции, вычислить:

- Синус числа *x*;
- Целую часть числа *x*;
- Результат округления числа *x* до целого;
- Арктангенс числа *x*;
- Результат округления числа *x* до двух десятичных знаков;
- Корень квадратный из числа *x*;
- Число *x* в кубе.

Все результаты вывести на экран.

2. Использование символьных функций.

Создайте программу prog17. В программе выполните следующие действия: переменной *str* присвойте произвольное строковое значение, длиной не менее 20 символов (например, 'Техническое задание').

Используя встроенные символьные функции, выполнить:

- Удалить все начальные и конечные пробелы из строки;
- Выделить из начала строки 5 символов;
- Определить номер позиции, с которой начинается подстрока 'а';
- Преобразовать все буквы в заглавные;
- Выделить из строки 5 символов, начиная с третьего;
- Вычислить длину строки.

Все результаты вывести на экран.

3. Использование функций по обработке даты.

Создайте программу prog18. В программе выполните следующие действия:

переменной `dd` присвойте значение системной даты.

Используя встроенные функции, выполнить:

- Выделить номер дня из даты;
- Выделить номер месяца из даты;
- Выделить номер года из даты;
- Все результаты вывести на экран.

4. Покажите результат преподавателю.

Контрольные вопросы:

1. Перечислите типы данных в БД.

Лабораторная работа 13

«Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям. Поиск данных в таблице».

Тема 5. Организация запросов SQL

Цель работы: «Получить некоторые практические навыки по созданию меню в программе Visual FoxPro»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7,

FoxPro **Краткие теоретические сведения:**

Команды для осуществления сортировки:

Sort to <имя новой БД> on <имя поля> — сортирует и создаёт новую базу данных с уже отсортированными данными

Ключи команды: [/A | /D] [/C]

[/A] — задает возрастающий порядок для данного поля

[/D] — убывающий порядок

SET DATE — Задает формат отображаемых выражений типа даты и DateTime.

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Сортировка базы данных Задание:

1. Отсортировать базу данных

1. Вызываем базу которую мы будем сортировать

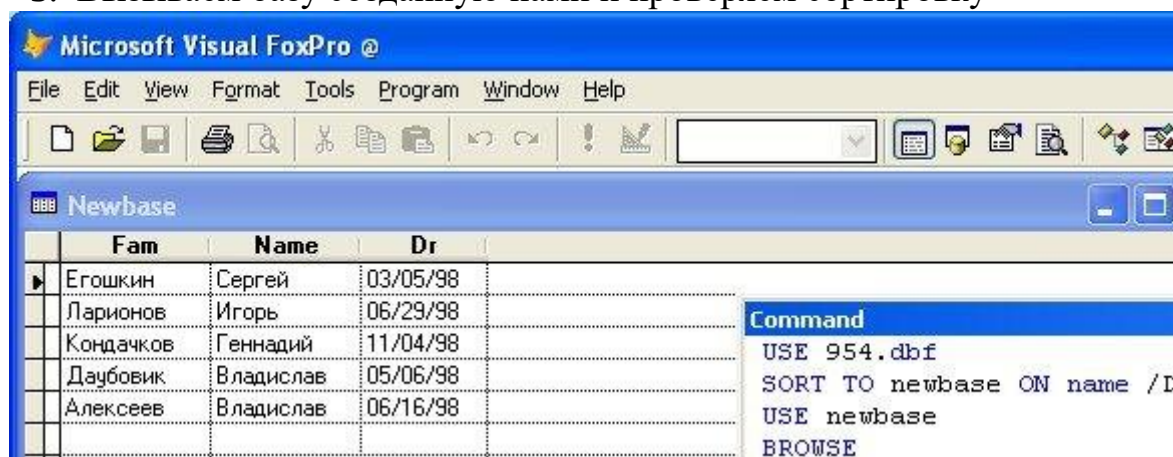


```
Command
USE 954.dbf
```

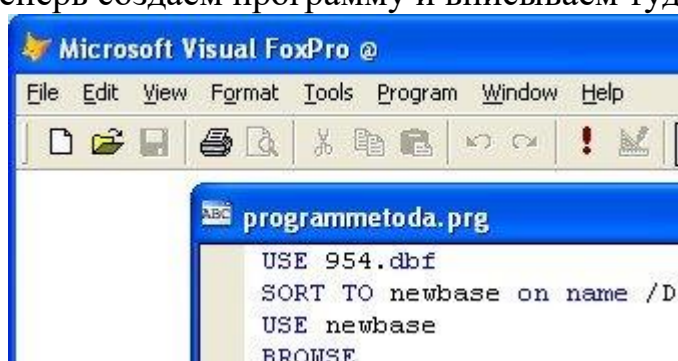
2. Сортируем базу 954.dbf по параметру “name” на убывание при этом создаём новую базу под названием “newbase”

```
Command
USE 954.dbf
SORT TO newbase ON name /D
```

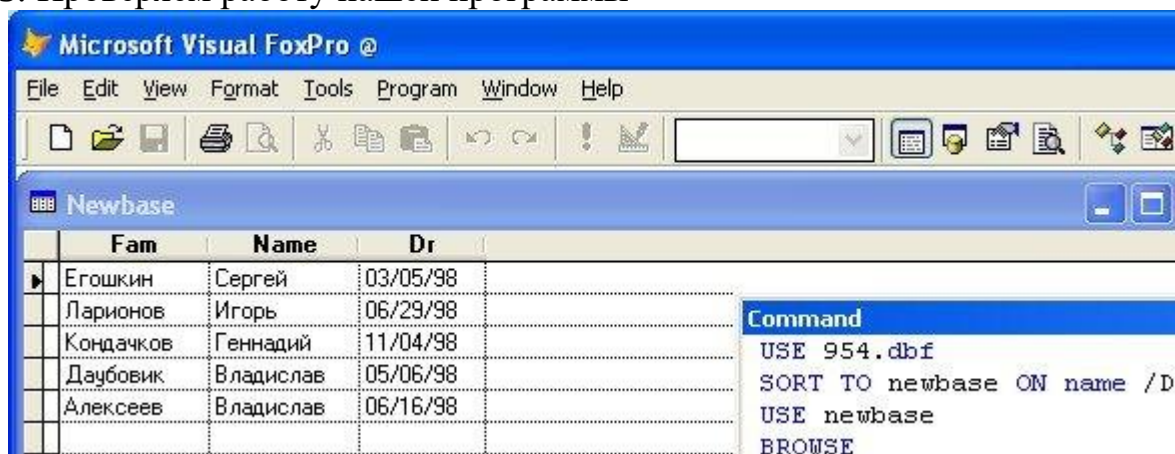
3. Вызываем базу созданную нами и проверяем сортировку



Как мы видим поле “name” прошло сортировку в порядке убывания. Теперь создаём программу и вписываем туда наши команды



5. Проверяем работу нашей программы



Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE

MENU <Название меню>

Создание поиска

Запускаем программу меню:

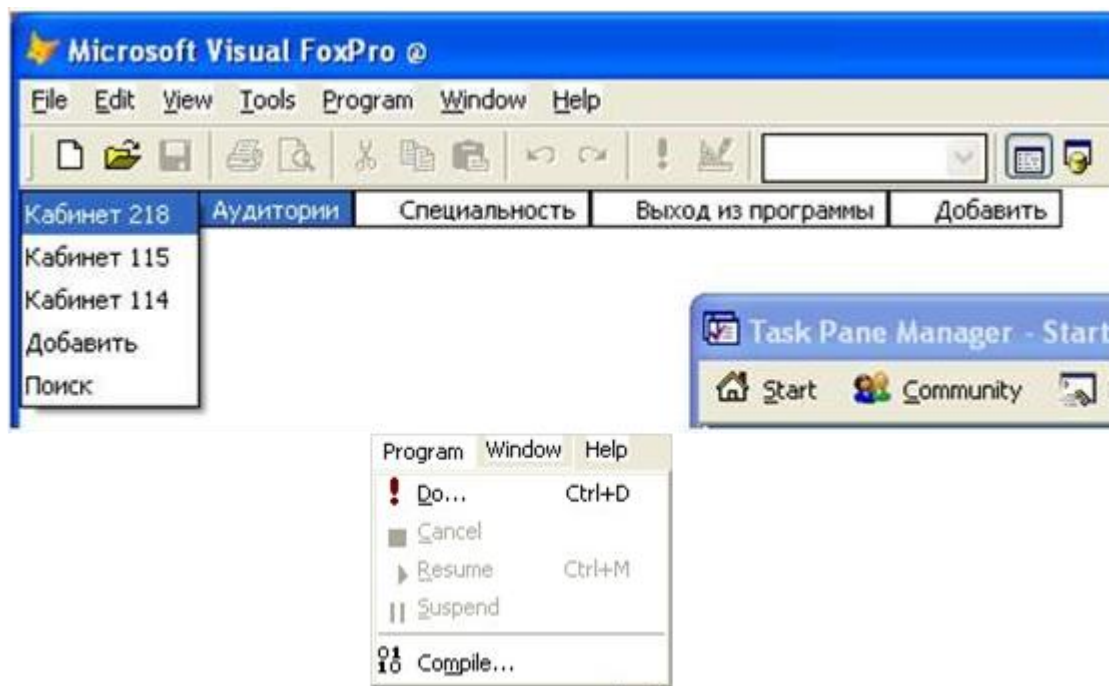


Рис.1(Наше меню.)

Далее нажимаем на кнопку «Поиск» дабы нам необходимо запустить нашу программу:

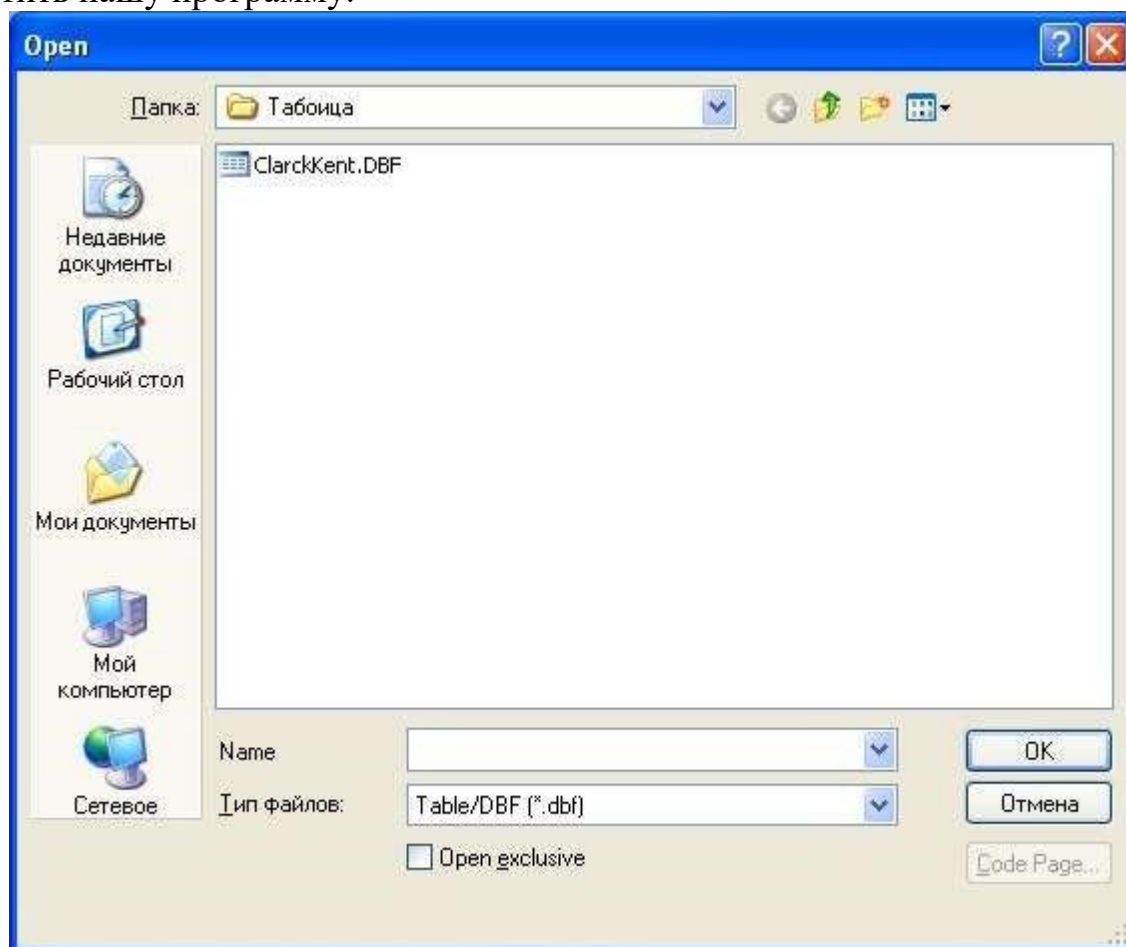


Рис.2(Открывается поиск нашей таблицы.) Выбираем.

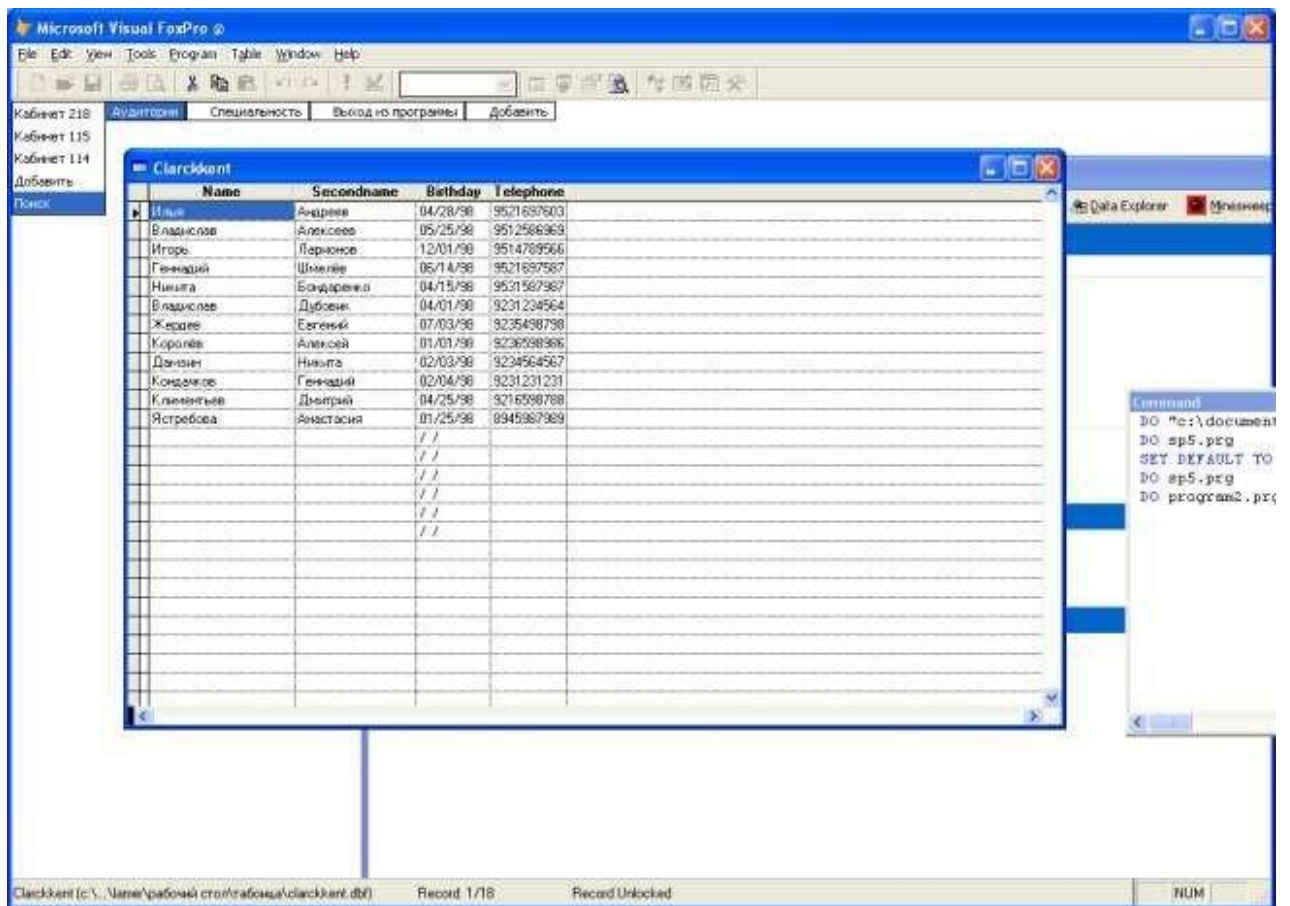


Рис.3(Открылась таблица.)

Заходим в пункт меню «Поиск»

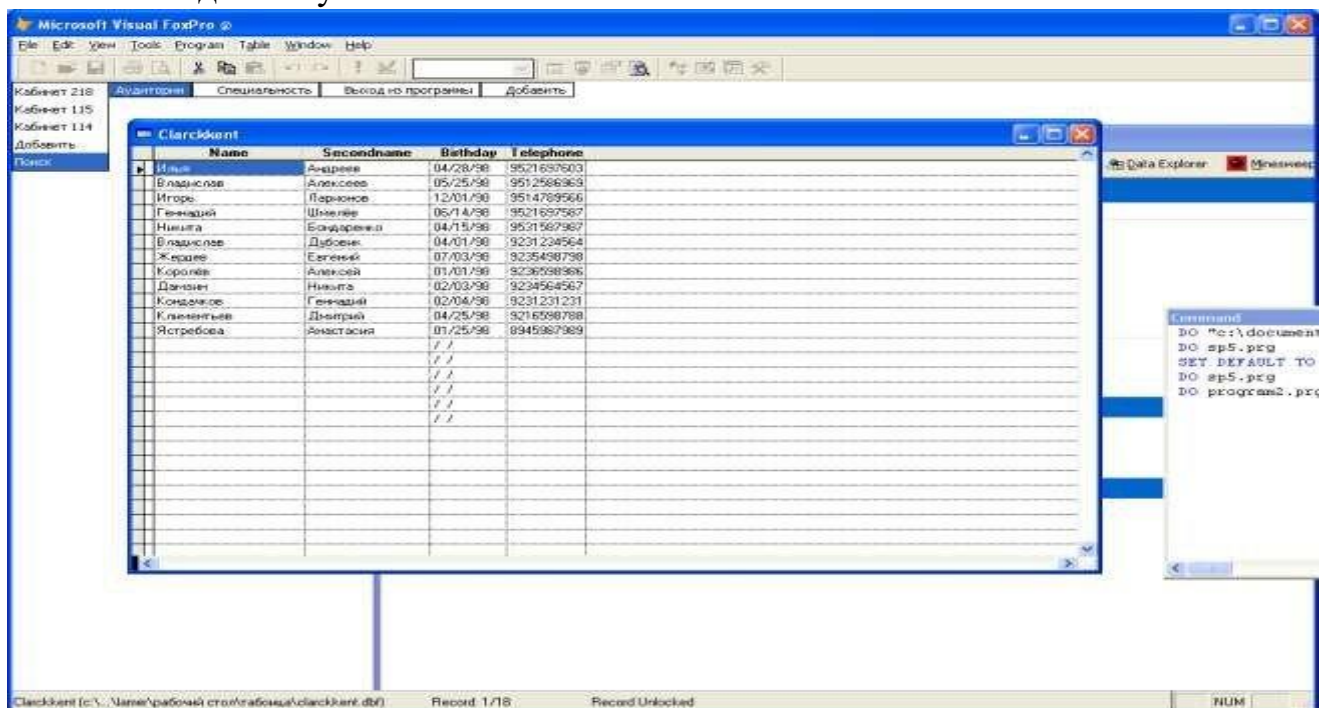


Рис.4 (Ну и соответственно видим нашу выполненную работу.) И немного расскажем, как мы это сделали:

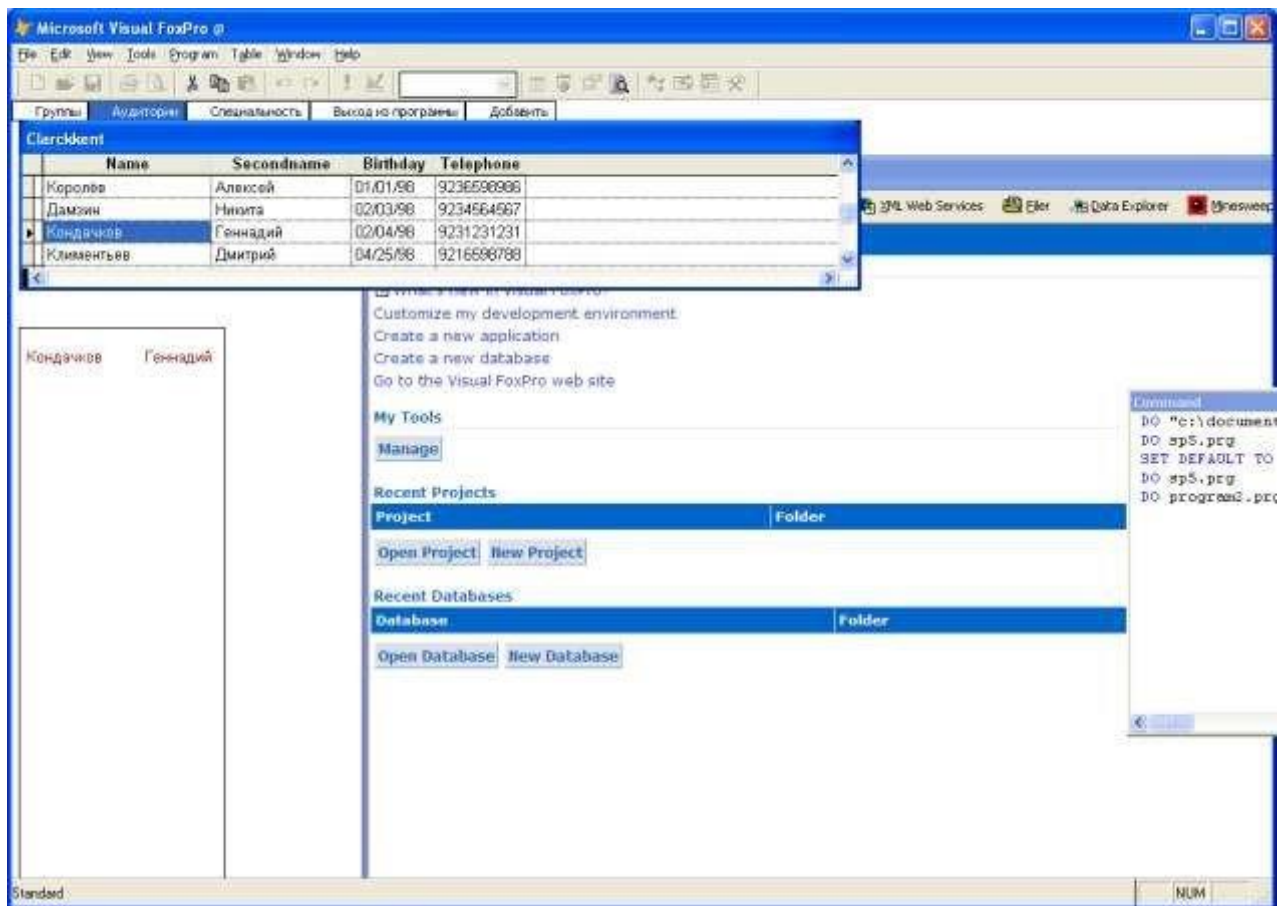


Рис.5(Добавление пункта)

Прописываем в нашем меню пункт «Поиск» чтобы он обращался к нашей программе, которая будет выполнять поиск.

Ну, и конечно же, наша сама программа выполняющая поиск:

```
define popup B
define bar 1 of B prompt "кабинет 218"
define bar 2 of B prompt "кабинет 115"
define bar 3 of B prompt "кабинет 114"
define bar 4 of B prompt "добавить"
define bar 5 of B prompt "поиск"
ON SELECTION PAD f2 OF Andreev activate popup B
on selection bar 1 of B do sp1
on selection bar 2 of B do sp1
on selection bar 3 of B do sp2
on selection bar 4 of B do sp4
on selection bar 5 of B do sp5
```

Рис.6(Программа)

Контрольные вопросы:

1. Для чего предназначена сортировка?
2. Как осуществляется сортировка в FoxPro?
3. Как осуществить поиск в БД?

Лабораторная работа 14

«Работа с переменными. Написание программного файла и работа с табличными файлами. Заполнение массива из табличного файла. Заполнение табличного файла из массива».

Тема 5. Организация запросов SQL

Цель работы: «Преобразование реляционной БД»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7,

FoxPro

Краткие теоретические сведения:

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы
 - и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

«Создание базы данных и заполнение таблиц»

1. Разработка структуры базы данных.

Для создания новой БД выберите пункт меню **File/New**, в окне **New** установите флажок **Database** и нажмите кнопку **New File**. Для сохранения БД укажите путь к своей папке и введите имя новой БД - Mydata. Откроется окно Дизайнера БД.



Для работы с базой данных используется панель инструментов Дизайнера БД.

Назначение кнопок:

- создать новую таблицу
- добавить существующую таблицу
- удалить таблицу из БД
- создать новое удаленное представление
- создать новое локальное представление
- изменить структуру и свойства таблицы
- просмотр таблицы
- редактирование хранимых процедур
- создание удаленной БД

2.Разработка структур таблиц базы данных.

Создайте в БД первую таблицу (Типы книг) с именем *tip*. Поля таблицы *tip*

Имя поля	Тип поля, размер	Описание (св-во Caption)
Kodt	Integer (AutoInc)	Код типа
Name	Character, 30	Наименование типа

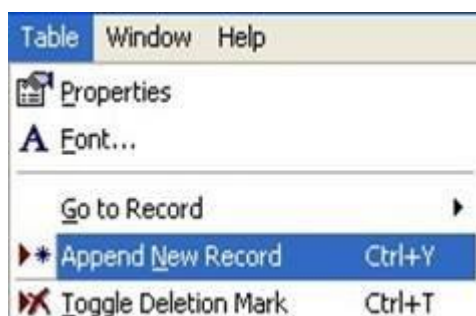
Создайте вторую таблицу (Книги) с именем *knig*. Поля таблицы *knig*

Имя поля	Тип поля, размер	Описание (св-во Caption)
Kodt	integer	Код типа
Kodkn	Integer (AutoInc)	Код книги
Autor	Character, 30	Автор
Nazvanie	Character, 40	Название
Cena	Numeric, 10,2	Цена
Kol	Integer	Количество

3. Ввод данных в таблицы.

Для добавления записей необходимо выбрать нужную таблицу и нажать на кнопку **Browse**.

Чтобы добавить пустую запись, нужно выбрать в меню **Table** команду **Append New Record**. Затем ввести данные в пустую запись.



Второй способ – набрать команду **Append Blank** в командном окне.

Введите в таблицу *Tip* следующие данные:

Kodt	Name
1	Художественная
2	Научная
3	Журналы
4	Газеты
5	Публицистическая

Введите в таблицу *knig* следующие данные:

Kodt	Kodkn	Autor	Nazvanie	Cena	Kol
1	1	Пушкин А.С.	Евгений Онегин	150.00	2
1	2	Есенин С.А.	Лирика	95.00	7
3	3		Мир ПК №1	50.00	5
3	4		Мир ПК №2	55.00	5
5	5	Герхард С.	Мир фантастики	100.00	4
2	6	Крылов Е.В.	Мы и закон	90.00	3
4	7		168 часов № 55	15.00	5
2	8	Иванов А.М.	Молекулярная физика	125.00	1
1	9	Кузнецов А.А.	Преступление века	199.00	2
5	10	Розенберг Е.С.	Закон добра	255.00	3
2	11	Прохоров В. И.	Жизнь	96.00	10
2	12	Лейман А.П.	Математический анализ	180.00	12
4	13		Комсомольская правда	25.00	2

4. Покажите результат преподавателю.

5. Завершите работу приложения.

4. Выполнение самостоятельного задания.

По аналогии с выполненным заданием создайте базу данных предложенного вам варианта.

Заполните созданные таблицы контрольными данными (введите новые записи) - в главной таблице не менее 5 записей, в дочерней – не менее 10.

«Ввод, редактирование и удаление данных» Ввод записей в таблицу.

Откройте в своей папке созданную ранее БД Mydata, которая содержит две таблицы – *Tip* и *Knig*.

Добавьте в таблицу **Tip** новую запись – поле kodt=6 , поле name= ‘Учебники’. Откройте таблицу **Knig**. Добавьте в нее дополнительно две произвольные записи по типу книг Учебники (kodt=6).

1. Редактирование данных в таблице.

Перейдите в таблицу **Tip**. Выполните в командном окне команду:

REPLACE name WITH 'Научно-фантастическая' FOR name='Научная'

Просмотрите результат замены данных в таблице – команда BROWSE.

Аналогично замените данные ‘Учебники’ на ‘Учебники и задачки’.

Выполненную команду и результат скопируйте и вставьте в файл отчета.

Перейдите в таблицу **Knig**. Выполните в командном окне следующие команды по замене данных:

- заменить поле nazvanie значением ‘Молекулярная химия’, в записи со значением этого поля ‘Химия’;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле kol значением 3 у всех записей со значением 2;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле cena значением 30.00 у всех записей со значением этого поля, меньшим 30;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле cena значением 20.00 у записи со значением поля nazvanie = ‘168 часов № 55’;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**

2. Удаление данных в таблице.

Перейдите в таблицу **Knig**. Выполните в командном окне команду:

DELETE ALL FOR kol<4

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет). Сколько записей было удалено с помощью этой команды?

Выполните команду:

RECALL all

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет). Что произошло в ранее удаленными записями?

Аналогичным образом выполните следующие команды по удалению данных:

- удалить все записи, у которых значение поля kodt равно 1;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля kol равно 5;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля сена больше 100;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля kol больше 8 и значение поля сена меньше 100;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите все удаленные записи.

3. Вывод данных на экран дисплея.

Перейдите в таблицу *Knig*. Установите указатель на запись, у которой поле kodkn=6.

Выполните в командном окне команду: DISPLAY

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).

Выполните в командном окне команду: DISPLAY ALL

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).

Используя аналогичным образом команды вывода данных, выполните следующие действия:

- Вывести на экран поля autor и nazvanie всех записей;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Вывести на экран поля autor, nazvanie и kol всех записей, у которых количество больше 9; • **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Вывести на экран поля autor, nazvanie и сена всех записей, у которых цена меньше 120;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**

- Вывести на экран поля *autor*, *pazvanie*, *sena* и *kol* всех записей, у которых поле *kodt=6*;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Вывести на экран структуру текущей таблицы;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**

Выполните аналогичные действия, используя команду LIST.

4. Перемещение по записям таблицы.

Перейдите в таблицу **Knig**. Используя команды перемещения, выполните следующие действия:

- переместите с помощью команды указатель на последнюю запись в таблице;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- **проверьте значение функции eof() (для вывода значения на экран выполните команду ? eof()); (дополните отчет);**
- переместите с помощью команды указатель на первую запись в таблице;
- **проверьте результат выполнения команды, просмотрев данные;**
- **проверьте значение функции bof() (для вывода значения на экран выполните команду ? bof()); (дополните отчет);**
- переместите с помощью команды указатель на запись с номером 5;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на предыдущую запись;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на 3 записи вниз;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на 4 записи вверх;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**

5. Покажите файл с результатами преподавателю.

6. Выполнение самостоятельного задания.

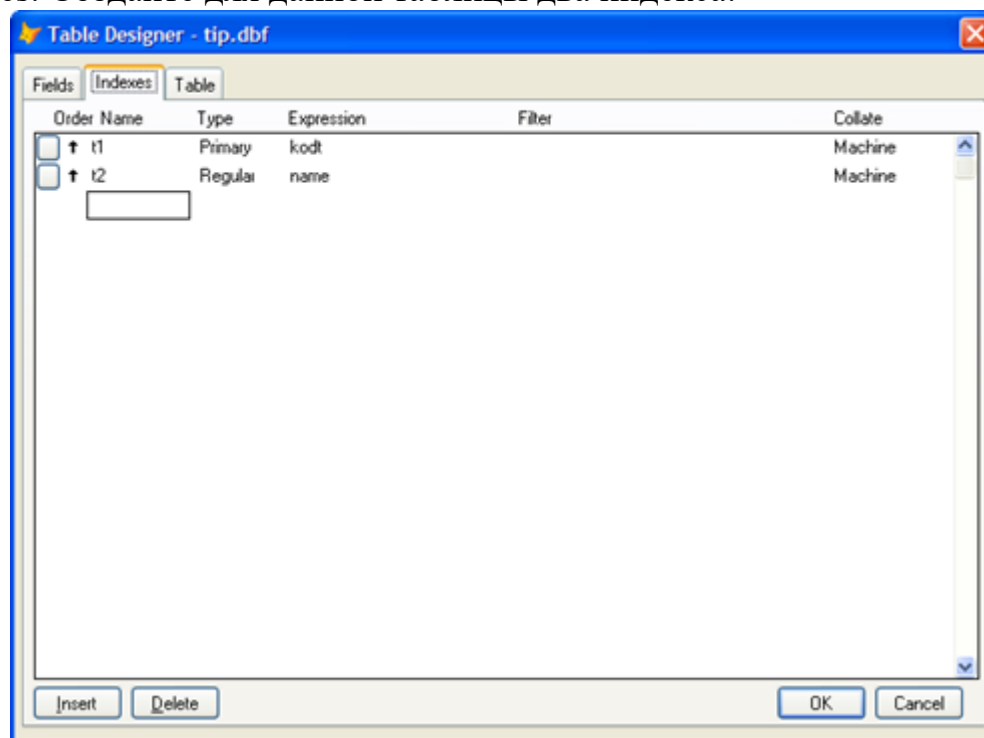
Откройте базу данных вашего варианта, созданную ранее. Добавьте в таблицы недостающие данные.

Выполните в командном окне тренировочные команды по замене данных, удалению и восстановлению записей, выводу записей на экран, перемещению по записям таблицы.

«Создание индексов и определение отношений между таблицами»

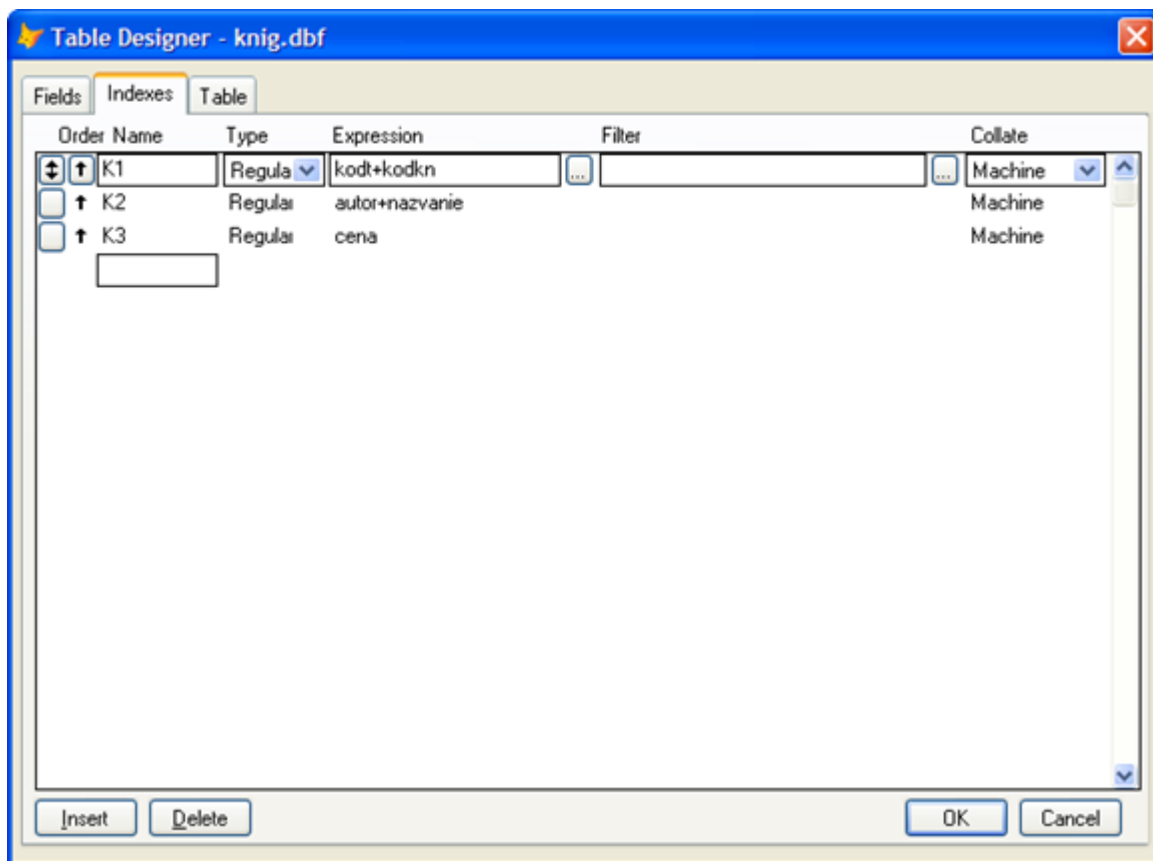
1. *Создание индексов с помощью конструктора.*

Откройте в своей папке созданную ранее БД Mydata. Для таблицы Tip откройте конструктор таблиц и в окне конструктора перейдите на вкладку Indexes. Создайте для данной таблицы два индекса:



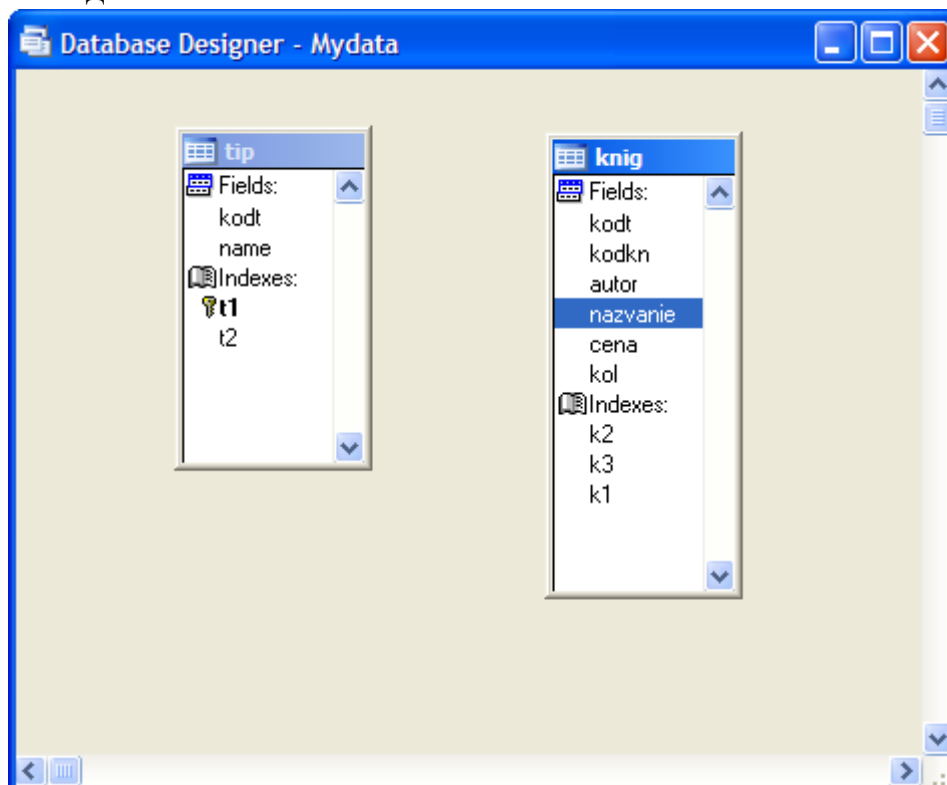
Сохраните внесенные изменения. Обратите внимание, что созданные индексы появились в макете таблицы в конструкторе БД.

Аналогичным образом создайте индексы для таблицы Knig:

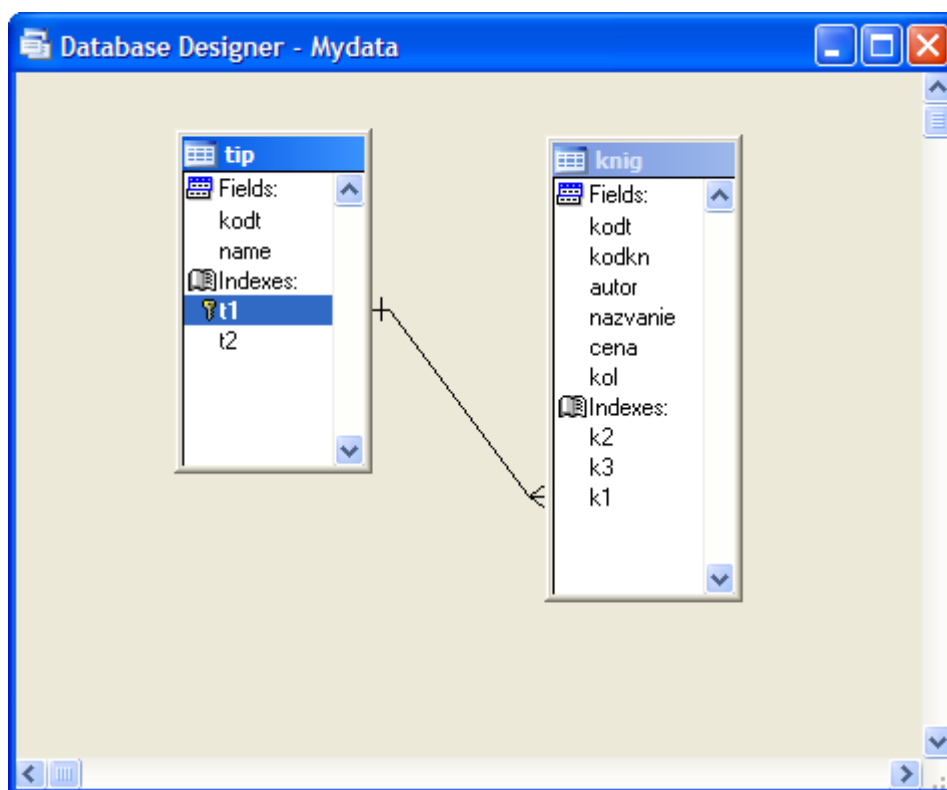


2. Установка отношений между таблицами.

Перейдите в конструктор БД, в котором находятся таблицы Tip и Knig, с созданными индексами.



Установите долговременную связь между таблицами, перетаскив ключевой индекс главной таблицы на соответствующий индекс дочерней.



3. **Контроль целостности данных**

Для созданной связи выберите пункт Edit Referential Integrity контекстного меню и установите каскадные правила обновления и каскадные правила удаления, а также правила вставки Restrict.

4. **Выполнение самостоятельного задания**

Откройте базу данных вашего варианта, созданную ранее. Создайте для каждой таблицы необходимые индексы. Для главной таблицы должен быть обязательно создан первичный (ключевой) индекс.

Установите долговременные отношения между таблицами, используя созданные индексы. Осуществите контроль целостности данных.

5. **Покажите результат преподавателю.**

«Использование индексов»

1. **Создание дополнительных индексов.**

Откройте в своей папке созданную ранее БД Mydata. Для таблицы Knig создайте дополнительные индексы:

- Индекс k4 - по полю kodkn;
- Индекс k5 - по полю kol;
- Индекс k6 - по полям autor и kol; 2. **Выбор активного индекса в процессе выполнения.**

Откройте таблицу Knig. Выполните команды, которые по очереди подключают все индексы (set order to ...).

После подключения очередного индекса, просмотрите результат и добавьте команду и результат в отчет.

3. Обновление индексов.

Откройте таблицу Knig в монопольном режиме. Выполните для данной таблицы переиндексирование.

Аналогично выполните обновление индексов для таблицы Tip. **4.**

Выполнение самостоятельного задания.

Откройте базу данных вашего варианта, созданную ранее. Создайте для каждой таблицы дополнительные индексы по разным полям.

Выполните подключение индексов в процессе работы и просмотрите результаты.

5. Покажите файл с результатами преподавателю.

Контрольные вопросы:

1. Что такое индекс?
2. Опишите этапы создания БД?
3. Перечислите основные элементы БД?

Лабораторная работа 15

«Добавление записей в табличный файл из двумерного массива. Работа с командами ввода-вывода. Использование функций для работы с массивами».

Тема 5. Организация запросов SQL

Цель работы: «создать, отредактировать запись в таблице»

Материально-техническое обеспечение: Компьютер, операционная система Windows 7, FoxPro

Краткие теоретические сведения:

Порядок выполнения лабораторной работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование лабораторной работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения лабораторной работы:

Добавление записи в таблицу через команды

1. Нам необходимо в первую очередь, конечно же открыть нашу Базу Данных.
2. Далее нам необходимо создать программу, через которую, будет все работать.
3. Далее необходимо наше «Меню» которое мы сделали в «3 работе по FoxPro». Оно нам пригодиться для того, чтобы добавить новый пункт, который нужен в нашей лабораторной работе.
4. Подготовить отчет.
5. Написать вывод о проделанной работе.

Для начала нам нужно открыть нашу Базу Данных:

Для этого вам необходимо написать команду в окне «Commands» команду «Browse» и нажать Enter:

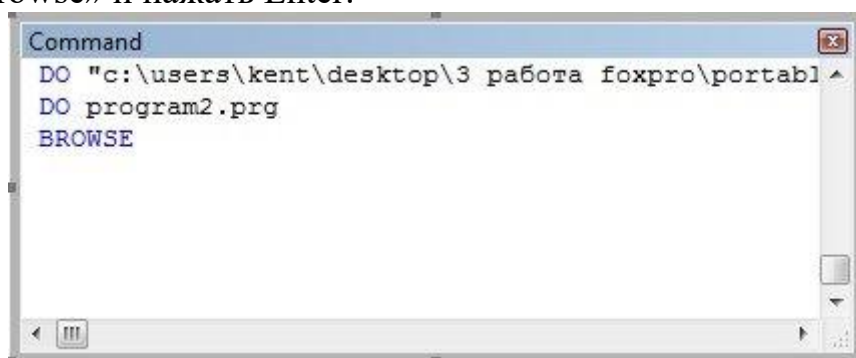


Рис.1(Команда Browse)

Далее у вас откроется окно, для выбора вашей Базы Данных:

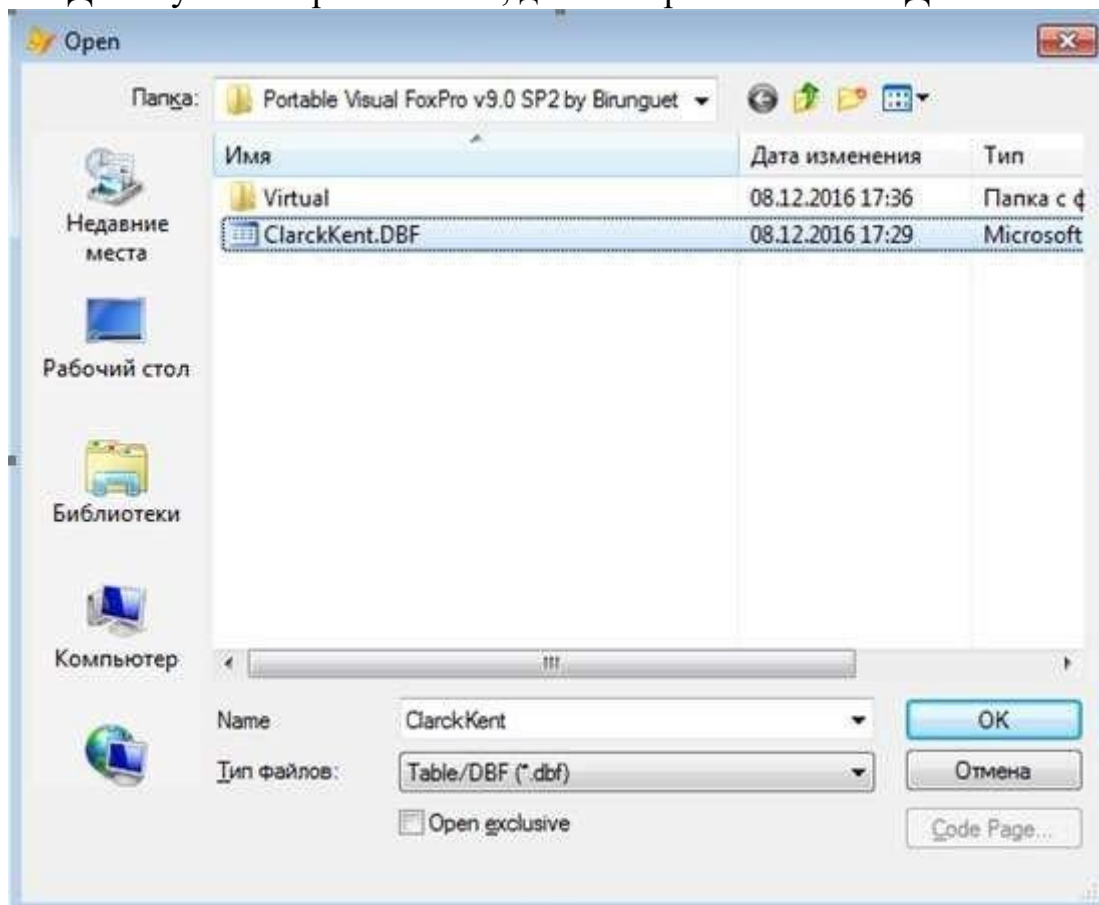


Рис.2(Выполнение команды «Browse»)

Нажимаем «ОК» и видим нашу таблицу с нашими данными:



	Name	Secondname	Birthday	Telephone
▶	Илья	Андреев	04/28/98	9521697603
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королев	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989

Рис.3(Наша таблица)

Чтобы создать новый файл, необходимо: New -> Programm -> New file:

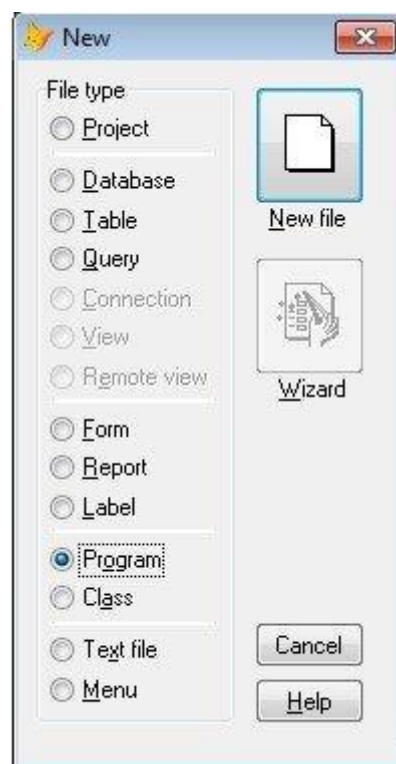


Рис.4(Создание)

После того, как мы создали программу, необходимо вписать эти параметры, у меня она будет называться «sr4».

Команда USE у каждого по-своему, она выбирает ваш путь, к вашей же таблице:

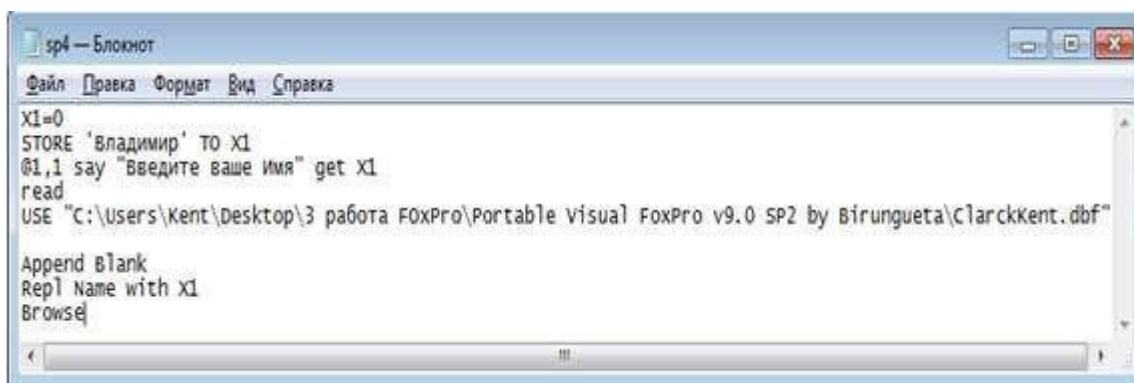


Рис.5(Programm)

Но после этого, не запускайте ее, потому что мы не определили ее в нашем меню.

Для того, чтобы добавить эту программу в само меню, вам необходимо проделать несколько действий:

Для начала зайти в программу, где находится сама меню.

Давайте добавим новый пункт меню, который будет носить имя «Добавить».

Сначала впишем ее, чтобы она отображалась: Я добавлю ее в выпадающую кнопку «Аудитории»:

```
define bar 1 of B prompt "кабинет 218"
define bar 2 of B prompt "кабинет 115"
define bar 3 of B prompt "кабинет 114"
define bar 4 of B prompt "добавить"
ON SELECTION PAD f2 OF Andreev activate popup B |
```

Рис.6(Кнопка «Добавить»)

Теперь сделаем, чтобы наша кнопка «Добавить» обращалась к нашей программе «sp4»:

```
on selection bar 1 of B do sp1
on selection bar 2 of B do sp1
on selection bar 3 of B do sp2
on selection bar 4 of B do sp4
```

Рис.7(Добавление программы для работоспособности)

И конечно же, проверим работоспособность, для этого:



Рис.8(Проверка)

После откроется окно, для того, чтобы вы выбрали свою программу:

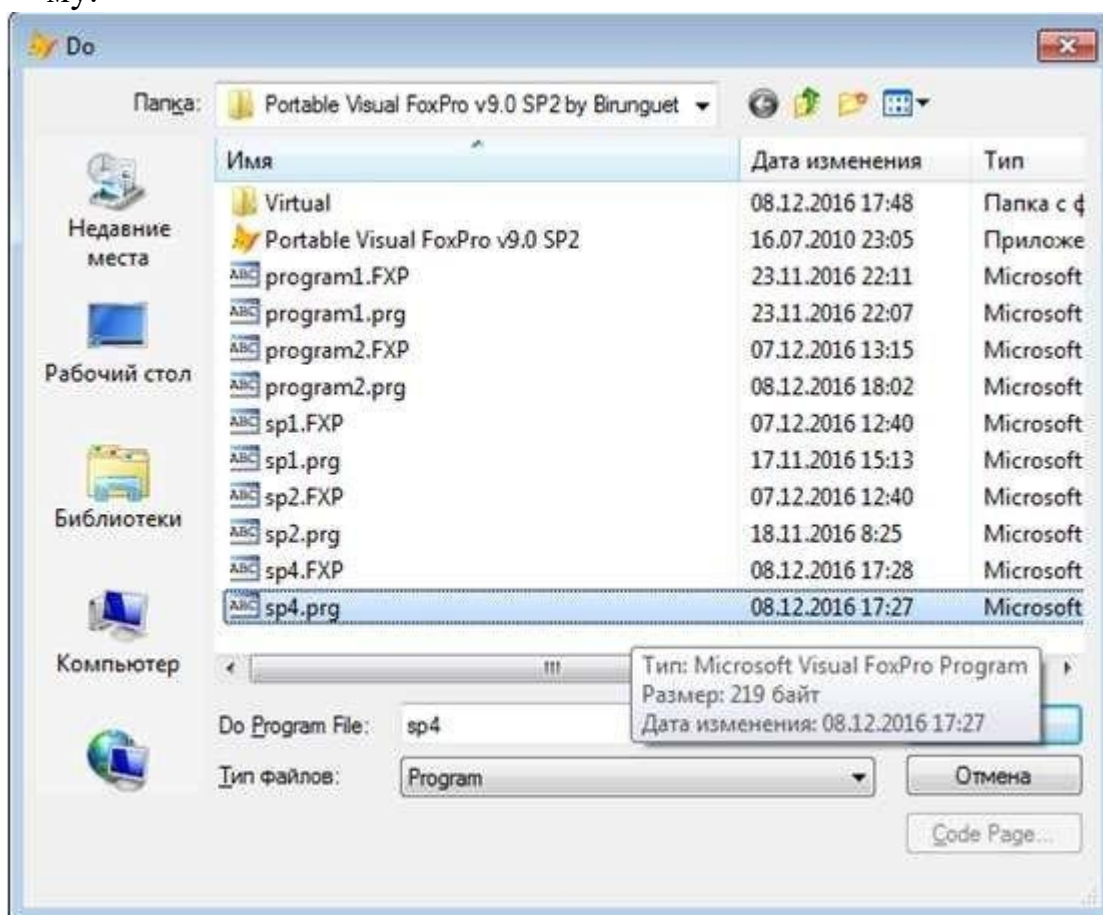


Рис.9(Выбор программы)

После выбора видим:

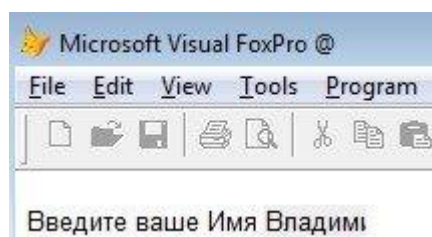


Рис.10(Выполнение программы)

Вам надо стереть то, что написано в поле, и ввести то, что нужно вам.

Ввели, и смотрим, что получилось:

Clarckkent				
	Name	Secondname	Birthday	Telephone
	Илья	Андреев	04/28/98	9521697603
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
►	Владимир		/ /	

Рис.11(Проверка работоспособности)

Контрольные вопросы:

1. Что такое запись в таблице?
2. Как можно добавить запись в таблицу?
3. Какие параметры можно настроить для записи?

Варианты заданий

Задание 1. Добавить в таблицу JOBS новую запись, указав при этом следующие значения: JOB_ID='HR_MAN', JOB_TITLE='Human Resources Manager' и MIN_SALARY=4500.

Задание 2. Добавить новую запись в таблицу JOB_HISTORY, указав при этом EMPLOYEE_ID=111, START_DATE=TO_DATE ('28.09.97'), END_DATE=TO_DATE('31.12.09').

Задание 3. Запрос на добавление данных о новом сотруднике построить так, чтобы для создаваемой учетной записи сотрудника в столбец SALARY автоматически добавлялось минимальное значение заработной платы для указанной должности.

Задание 4. Запишите директиву языка SQL для добавления новой записи в таблицу COUNTRIES без указания списка столбцов этой таблицы.

Задание 5. Добавить в таблицу EMP_TEMP данные о сотрудниках, которые занимаются закупками товаров (Purchasing) и их продажей (Sales).

Задание 6. Используя в качестве источника записей соединение таблиц LOCATIONS и COUNTRIES, заполнить с помощью единственного оператора INSERT таблицы LOCS_1, LOCS_2 и LOCS_3, поместив туда данные, которые относятся к Европе (REGION_ID=1), Америке (REGION_ID=2) и Азии (REGION_ID=3), соответственно. Структуру таблиц LOCS_1, LOCS_2 и LOCS_3 выбрать по своему усмотрению.

При выполнении этого задания рекомендуется сначала воспользоваться следующим оператором для создания новой таблицы:

```
CREATE TABLE LOCS_1  
AS SELECT * FROM LOCATIONS NATURAL JOIN COUNTRIES  
WHERE 1=2
```

После выполнения этого оператора появится пустая таблица LOCS_1, структура которой будет объединять столбцы исходных таблиц LOCATIONS и COUNTRIES. Теперь остается вставить в нее требуемые записи. Одновременно заметим, что если в подзапросе, который применяется для создания новой таблицы, вместо символа * указать только некоторые столбцы исходных таблиц, то новая таблица будет содержать только эти столбцы.

Задание 7. Повысить оклад на 10% всем руководителям департаментов.

Задание 8. Сотрудника, у которого LAST_NAME='Kozlov', перевести в департамент Accounting на должность Accounting Manager и установить ему зарплату в размере 8500\$.

Задание 9. Сотрудника, у которого EMPLOYEE_ID=300, перевести в подчинение к тому же менеджеру, как у EMPLOYEE_ID=105, и установить зарплату на 10% выше минимума для занимаемой должности.

Задание 10. В таблице EMP_TEMP оставить только данные о сотрудниках, которые занимаются продажами (Sales) и маркетингом (Marketing).

Задание 11. В таблице EMP_LOW оставить только данные о сотрудниках, у которых заработная плата ниже половины средней зарплаты по компании.

4. ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Что обозначает аббревиатура SQL?
2. Будет ли правильным утверждать, что SQL является непроцедурным языком?
3. На какие категории разделяются команды языка SQL?
4. Какие функции выполняют команды управления транзакциями?
5. Перечислите основные предложения оператора SELECT. Какие из них являются обязательными?
6. Проанализируйте корректность следующих запросов:
 - a) Select *
 - b) Select * from checks
 - c) Select amount name payee FROM checks.
7. Какая из агрегатных функций SUM, COUNT, MIN, MAX, AVG возвращает множество значений?
8. Будут ли корректными следующие утверждения:
 - a) степень вложенности подзапросов не может превышать 2;
 - b) коррелированные подзапросы являются полностью самостоятельными (независимыми).
9. Можно ли в выражении для ключевого слова WHERE задать несколько условий?
10. Будет ли правильным утверждение, что при использовании ключевого слова IN проверяемое значение должно совпадать с каждым элементом списка?
11. Будет ли правильным утверждение, что при наличии ключевого слова NAVING необходимо также использовать ключевые слова GROUP BY?
12. Какой тип соединения таблиц необходимо использовать в запросе, чтобы извлечь записи из некоторой таблицы независимо от наличия связанных записей в другой таблице?
13. В чем состоит особенность натурального (естественного) соединения двух таблиц?
14. Каким будет результат, если при выборке из двух таблиц не указать условие для связывания их записей?
15. Какие разделы могут использоваться при составлении команды SELECT для выбора данных из таблицы?
16. Какая директива языка SQL используется для создания таблиц?
17. Какие основные типы данных могут использоваться при создании таблиц?
18. Что такое псевдоним (алиас) и каким образом он определяется?
19. Какой формат имеет SQL-команда для уничтожения таблицы?
20. Поясните форматы SQL-команды для вставки новых записей в таблицу.
21. Каким образом можно указать конкретные колонки таблицы в запросе на выборку данных?

22. Каким образом оператор SELECT позволяет осуществить выбор строк таблицы, удовлетворяющих заданным условиям?

23. Какие специальные символы могут использоваться в функции LIKE для сравнения по образцу?

24. Какое ключевое слово задает режим запрета вывода строк-дубликатов?

25. Какими способами в операторе SELECT можно задать условия для соединения таблиц?

26. Какие объекты базы данных обеспечивают функционирование старых приложений без их модификации в случае изменения структуры БД?

27. Можно ли создать представление (view), которое будет работать одновременно с несколькими таблицами БД?

28. Каким образом можно перенести (по заданному условию) данные из одной таблицы в другую?

29. Каким целям служат представления? Чем они функционально отличаются от таблиц?

30. Как NULL-значения влияют на результаты арифметических операций?

31. Как правильно выполнять сравнение с NULL-значением?

32. Перечислите основные агрегатные функции и правила их использования.

33. Можно ли выполнить запрос на выборку, если данные расположены в нескольких таблицах?

34. В чём разница между INNER JOIN и OUTER JOIN?

35. Объясните отличие в использовании предложений WHERE и HAVING.

36. Укажите причины, по которым операция добавления в таблицу новой записи может завершиться с ошибкой.

37. Можно ли одной командой добавить в таблицу несколько новых строк?

38. Как быстро очистить всю таблицу? В чём недостаток этой операции?

39. Укажите реляционную операцию (UNION, UNION ALL, INTERSECT, MINUS) для работы с результатами двух запросов на выборку, если требуется:

а) показать совпадающие записи;

б) показать все данные;

в) показать все данные, но без повторений;

г) показать только те строки первого запроса, которые не выдаются вторым.

40. Поясните, в чём разница между командами DROP TABLE TAB_1 и DELETE FROM TAB_1.

5. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основная учебная литература

1. Илюшечкин В. М. Основы использования и проектирования баз данных : учебник для СПО / В. М. Илюшечкин. — испр. и доп. — Москва : Издательство Юрайт, 2021. — 213 с. — (Профессиональное образование). — URL: <https://urait.ru/bcode/471698> .

2. Советов Б. Я. Базы данных : учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 420 с. — (Профессиональное образование). — URL: <https://urait.ru/bcode/472497>.

3. Шустова Л. И. Базы данных : учебник / Л. И. Шустова, О. В. Тараканов. — Москва : ИНФРА-М, 2021. — 304 с. — (Среднее профессиональное образование).

Дополнительная учебная литература

4. Голицына О. Л. Основы проектирования баз данных : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. — 2-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 416 с. — (Среднее профессиональное образование). — URL: <https://znanium.com/catalog/document?id=3649005>.

5. Стружкин Н. П. Базы данных: проектирование. Практикум : учебное пособие для СПО / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2021. — 291 с. — (Профессиональное образование).

6. Федорова Г. Н. Основы проектирования баз данных: учебник / Г. Н. Федорова. — Москва : ИЦ «Академия», 2020. — 224 с. - (Профессиональное образование).

6. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ РЕСУРСОВ

- Интернет-университет информационных технологий: www.intuit.ru.
- Портал Центра Информационных Технологий: www.citforum.ru.
- Электронная библиотечная система издательства «ИНФРА-М»: www.znanium.com.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

№ п/п	Темы дисциплины	Перечень учебно-методических материалов
1	Основы построения баз данных	1, 2, 3
2	Технология работы с данными в СУБД MS Access	4, 5, 6
3	Публикация данных в Internet и Intranet	4, 5, 6
4	Использование Access совместно с другими приложениями MS Office	4, 5, 6
5	Язык SQL и СУБД типа OracleXE 10G	1, 6