

# 11 Лабораторная работа № 2

## Практическое занятие № 1

### Создание запроса с вычисляемым полем

1. Создать новую папку в своем каталоге.
2. Открыть СУБД VISUAL- FOXPRO 9.0.
3. Создать базу данных «Склад».
3. В базе данных «Склад». Создать таблицу «Деталь» и заполнить ее следующей информацией:

Номер_склада	Шифр_детали	Количество	Стоимость	Дата поступления	Процент металла
5	X010	100	5.00	1.12.04	0.3
10	T010	200	100.50	5.11.04	9.5
10	Y030	5	200.00	3.12.04	0.005
17	X010	500	8.00	5.01.05	0.5
17	A050	8	150.00	15.03.05	11
20	B050	10	200.00	4.04.05	25
25	A010	12	120.78	23.03.05	40
25	Y010	30	130.48	2.03.05	25
10	S010	20	150.56	12.03.05	10.1

5. Составить запрос, используя, Мастер запросов, включив все поля таблицы «Деталь».

6. В Конструкторе запроса составить «вычисляемое поле - цена», для этого:

- в QUERY DESIGNER двойным щелчком мыши активизировать окно FUNCTION and EXPRESSION;
- в открывшемся окне набрать выражения КОЛИЧЕСТВО\*СТОИМОСТЬ AS ЦЕНА;
- далее следует выбрать ОК и вернуться конструктор запросов;
- в конструкторе запросов добавить новое поле;
- выполнить запрос результат просмотреть.
- сохранить результат в виде свободной таблице «Цена1»
- составить отчет по таблице «Цена1», сосчитать общую сумму.

## Работа № 1

### Создание меню

#### 1. Прочитать краткую справку о понятии МЕНЮ.

Цель создания меню - обеспечить пользователю простой доступ ко всем компонентам приложения. Обычно проект содержит свое собственное, которое заменяет основное меню VISUAL-FOXPRO и команды выполнения конкретных задач. С помощью меню организуется одновременная работа с такими заранее созданными объектами, как база данных, формы, отчеты, программы. Меню любой сложности строится из двух видов: горизонтальное и вертикальное (всплывающее). Горизонтальное меню, примером которого является основное меню VISUAL-FOXPRO состоит из нескольких расположенных пунктов, которые называются BAR и появляются только при активизации соответствующего PAD- пункта (например, пункты NEW, .OPEN, SAVE. пункта FILE основного меню).

#### 2. Создание меню визуальными средствами.

Для создания нового меню нужно выполнить команду FILE-NEW MENU. Диалоговое окно NEW MENU открывает два варианта меню:

1. MENU- линейное BAR-меню
- 2.SHORTACT- всплывающее POPUP-меню.

Выбрать MENU- CANCEL. Далее откроется окно MENU DESIGNER- MENU. В столбце PROMT вводится название пункта меню. Для разработанной базы данных введите, например, **1справочники**. В столбце RESULT выбираем один из предлагаемых вариантов, чтобы определить реакцию данного пункта на выбор:

**COMMAND** - выполняется команда, которая записывается правее;

**SUBMENU** - организуется всплывающее меню, для определения его пунктов нажмите кнопку CREATE справа;

**PROCEDURE** - выполняется процедура для ввода кода которой необходимо нажать кнопку EDIT;

**Pad NAME** или **Bar#** - используются элементы системного меню вводом одной из стандартных команд.

Для нашего случая в столбце RESULT выбрать SUBMENU. Для второго пункта меню в столбце PROMT набрать **2выход**, в поле RESULT - **COMMAND** и рядом в появившемся окне набрать команду **set sysmenu to**

**defa** (команда для восстановления системного меню).

Далее для пункта **1 справочник** создадим следующий уровень меню. Для этого вернемся на строку **1 справочник**, щелчком по кнопке CREATE создадим подменю. В столбце PROMT наберем Список, в столбце RESULT - COMMAND. Рядом в окне набрать команду **do form spisok.scx**, где **spisok.scx** - это имя файла, где находится форма, с которой вы работаете, у каждого имя может быть своим. Команда **do form spisok. scx** производит выполнение формы.

После создания меню в строке главного меню выбрать команду MENU-GENERATE. Сохранить созданное меню в своей папке, где находится ваша база данных. Далее активизировать GENERATE. Затем в строке главного меню выбрать команду PROGRAM - DO.

Выбрать файл с именем созданного меню с расширением .mpr и затем DO. В результате выполненных операций строка главного меню заменится на меню пользователя. Для выхода из разработанного меню нажать клавишу Выход.

## Работа 2

### Организация межтабличных связей в среде СУБД VISUAL FOXPRO 9.0

1. Изучить теоретический материал по индексированию таблиц.

#### 1.1. Понятие индексирования таблиц.

Просматривать и отыскивать данные в таблицах удобнее и быстрее тогда, когда они отсортированы (упорядочены) по одному или нескольким полям. Вводить данные в таблицу упорядоченными хотя бы по одному из полей проблемно. Поэтому данные вводятся в таблицы по мере необходимости и в случайном порядке, а сортировка данных при работе с ними осуществляется с помощью механизма индексирования таблиц.

Под **индексом** (индексным выражением) понимается имя поля таблицы или выражение, включающее совокупность имен полей, по которым упорядочена таблица. Индексирование таблиц позволяет осуществить быстрый поиск записей. Механизм индексирования базируется на использовании специального индексного файла, содержащего упорядоченные указатели(ссылки) на записи исходной таблицы, позволяющие извлекать записи в нужном порядке. Поэтому размер индексного файла гораздо меньше размера исходной таблицы. Индексный файл имеет имя совпадающее с именем таблицы и расширение CDX. Открываются и закрываются вместе с таблицами. Важное свойство индексов является возможность их использования для организации первичного ключа в таблице. Такие индексы должны быть уникальны, т. е. однозначно определять записи таблицы. Если ключевой индекс состоит из одного поля, то не должно быть двух записей в таблице с одинаковым значением этого поля. Для составного индекса, состоящего из нескольких имен полей, уникальным должна быть вся совокупность значений полей.

В VISUAL- FOXPRO 8.0 тип индексов задается при их создании с

помощью Конструктора таблиц - TABLE DESIGNER. В диалогом окне TABLE DESIGNER выбрать вкладку INDEXES (индексы). В поле NAME (имя) имя тега (индекса). Слева от имени индекса располагается переключатель, задающий направление упорядочивания значений индексного выражения (по возрастанию или убыванию значений). В поле TYPE (тип) выберем из списка подходящий тип для установления индекса.

**REGULAR** - значение индексного выражения записывается для каждой записи таблицы. При наличии одного и того же значения для нескольких записей в индексном файле будет указатель на каждую из них. При просмотре таблицы такие записи появляются в порядке их ввода.

**PRIMARY** - один из индексов, удовлетворяющий требованиям индекса типа CANDIDATE, может быть выбран в качестве первичного ключа. Используется для связывания таблиц.

**UNIQUE** - значение индексного выражения записывается только для первой из повторяющихся записей и только на нее в индексном файле есть указатель. При просмотре таблицы видна только одна (первая) из записей с одинаковым значением индексного выражения.

**CANDIDATE** - создается уникальный индекс не содержащий полей с пустыми значениями. Он является кандидатом на роль первичного ключа, но не является таковым, т. к. в таблице может быть только один первичный ключ.

## 1.2 Организация межтабличных связей

В VISUAL-FOXPRO возможно связывать таблицы и устанавливать связи между ними. При этом возможно установление постоянных и временных связей между таблицами. Постоянные связи можно установить в Конструкторе БАЗЫ ДАННЫХ. Временные связи между таблицами могут устанавливаться при создании экранных форм (для ввода редактирования данных) и отчетов. Обычно это делается в случаях, когда постоянные связи по каким-либо причинам не подходят для создаваемой формы или отчета. Постоянные связи хранятся в словаре базы данных и автоматически устанавливаются при открытии таблиц.

Условия установления постоянных связей:

1. Одна из связываемых таблиц является родительской, а другая дочерней.
2. Для родительской таблицы индекс должен быть определен как первичный ключ.
3. Для дочерней таблицы индекс может быть не уникальным. По этому индексу дочерняя таблица будет связываться с родительской.

Для установления постоянной связи между родительской и дочерней таблицей необходимо:

1. Установить указатель мыши на первичный ключ родительской таблицы.
2. Нажав левую кнопку мыши переместить указатель на индекс дочерней таблицы.
3. Отпустить мышь.

## 1.3. Организовать связь между двумя таблицами

1. Открыть VISUAL-FOXPRO
2. В меню FILE команду OPEN
3. Найти свой каталог, свою папку, где была сохранена база данных

## SCHOOL

4. Открыть файл SCHOOL.dbc
5. В результате выполнения операций откроется DATABASE DESIGNER- SCHOOL и созданная таблица SPISOK.
6. В окне Конструктора Базы данных установим курсор на таблицу, щелчком правой кнопки мыши и вызовом контекстного меню, затем выполнив команду MODIFY (модифицировать).
7. Установим на поле КОД индекс по возрастанию. Затем выберем вкладку INDEXES и зададим индекс поля КОД типа - PRIMARY; индекс поля ГРУППА - REGULAR.
8. Далее создадим новую таблицу GR с полями: группа - NUMERIC; преподаватель - CHARACTER.
9. В качестве ключа выберем поле группа тип индекса - PRIMARY.
10. Установим связь между таблицей SPISOK и GR. Для этого указатель мыши поместим на первичный ключ таблицы GR - группа и нажав левую кнопку мыши переместим указатель на индекс группы в таблице SPISOK. В результате проделанных операций установится связь 1: M.
11. Таблицу GR заполним следующими значениями в режиме BROWSE

<u>группа</u>	<u>преподаватель</u>
101	Верзаков
102	Белоусов
103	Масалова
104	Зачесова

## Практическое занятие № 2

### ДОБАВЛЕНИЕ ЗАПИСЕЙ В ФОРМЕ ПОЛЬЗОВАТЕЛЯ

1. Открыть созданную на предыдущем занятии форму и на поле формы нанести кнопку COMMAND BUTTON.
2. Изменить свойство CAPTION - добавить.
3. Два раза щелкнуть мышкой по добавленному объекту и набрать следующий программный код:

```
Select spisok
Set order to tag код
Kodn= код+1
append blank
replace код with kodn
form1.refresh
```

spisok - это имя таблицы на основе которой была построена форма;

form1- это имя формы с которой Вы работаете;

Select - возвращает номер текущей выбранной области;

Set - определяет индексный файл, который будет использоваться для таблицы;

tag - возвращает имена индексных файлов;

go bott - указатель записи на последней логической записи таблицы;

append blank- добавляет пустую запись в конец таблицы;

replace- заменяет данные в полях таблицы.

### **Практическая работа № 3** **Создание проекта приложения**

При создании приложения используется проект, который объединяет элементы приложения Visual FoxPro и группирует их по типам. Информация о проекте хранится в специальной таблице, которая, в отличие от обычных таблиц Visual FoxPro, имеет расширение PJX. Использование проекта упрощает разработку приложения, так как в проекте базы данных программы, формы, отчеты, запросы и другие элементы приложения располагаются в соответствующих разделах, а также запоминается расположение каждого включенного в проект элемента.

Создав проект и определив входящие в него элементы, вы можете использовать его для сборки приложения, построив файл с расширением APP, для создания исполняемого файла с расширением EXE. При построении приложения из проекта осуществляется поиск и сборка файлов, на которые ссылаются элементы приложения, отслеживаются версии файлов, входящих в проект.

Приступая к разработке нового приложения, прежде всего, создайте проект приложения. В дальнейшем вы будете добавлять в него созданные вами элементы приложения. Для создания нового проекта вы можете использовать мастер Application Wizard (Мастер приложения) или команду New (Новый) из меню File (Файл).

При выполнении команды New (Новый) на экране открывается соответствующее диалоговое окно с перечислением всех типов элементов приложения, которые возможны в Visual FoxPro. Установите опцию Project (Проект). Выполните следующие действия:

1. Нажмите кнопку New file (Новый файл).
2. В поле ввода Enter project file (Введите имя проекта) диалогового окна Create (Создать) задайте имя создаваемого проекта, убедившись, что в поле Тип файла установлен тип сохраняемого файла Project (Проект), а в поле Папка правильно выбрана папка, в которой вы хотите расположить свой проект. Для сохранения созданного проекта нажмите кнопку Сохранить. Visual FoxPro создаст файлы проекта и запишет их в

указанное место.

3. После этого откроется окно проекта Project Manager (Менеджер проекта) В верхней части окна проекта находятся вкладки, которые управляют отображением определенных типов файлов проекта, значительно облегчая разработку приложения. Вкладки окна Project Manager:

All(Все) - Все файлы.

Data (Данные) - Базы данных, таблицы, запросы, представления данных, хранимые процедуры.

Documents (Документы) - Формы, отчеты и этикетки.

Classes (Классы) – Классы.

Code (Коды) - Программы и библиотеки.

Other (Остальные) - Меню, файлы переменных памяти, клавишные макрокоманды.

В Visual FoxPro используется иерархическое отображение элементов проекта. На самом верхнем уровне находятся категории элементов проекта.

Data (Данные), Documents (Документы), Class Libraries (Библиотеки классов), Code (Исходный код) и Other (Другие).

Каждый иерархический список может находиться в свернутом или развернутом состоянии. Если элемент иерархического списка содержит элементы следующего уровня, то в свернутом состоянии с левой стороны наименования элемента отображается знак плюс. Для раскрытия списка элементов следующего уровня необходимо щелкнуть мышью на знаке плюс. При раскрытии списка знак плюс изменяется на знак минус. Например, иерархический список Data (Данные), имеет глубину в четыре уровня. Для свертывания вложенного списка элементов необходимо щелкнуть мышью на знаке минус.

Для базы данных SCHOOL сформировать структуру проекта, включить: базу данных, форму, меню.

Например, для того чтобы включить в проект объект базу данных нужно мышью щелкнуть по +DATABASE-ADD (команда на Project Manager) - далее выделяем нужную базу данных в открывшемся окне - ОК. Остальные объекты добавляем в базу данных аналогично.

4. Первый программный файл добавленный к проекту является главным по умолчанию. Если необходимо назначить другой программный файл в качестве главного, следует его выбрать и выполнить команду SET MAIN. Для нашего случая главным файлом сделаем файл, где хранится меню для работы с базой данных SCHOOL. В иерархической структуре объектов выбрать нужное меню (файл, где содержится нужный объект) и правой кнопкой мыши щелкнуть по нему, далее выбрать команду SET MAIN.

5. Следующий шаг построение приложения.

Выбираем кнопку BUIL...- Application- ок- имя приложения- сохранить.

6. Теперь при работе с базой данных нужно в командной строке выбрать команду PROGRAM- имя приложения (со значком VISUAL

FoxPro (JIICA))-DO. В результате выполнения команды появится строка меню по работе с разработанной базой данных.

## **Практическая работа № 4**

### **Создание, модификация, работа с базой данных в среде VISUAL-FOXPRO средствами языка SQL**

Язык **SQL** является инструментом для выборки и обработки информации, содержащейся в базе данных. **SQL** - это сокращенное название структурированного языка запросов. **SQL**- это язык программирования, который применяется для организации взаимодействия пользователя с базой данных. **SQL** - работает только с базой данных реляционного типа. Основной синтаксической единицей языка является инструкция, в языке существует около 49 инструкций, описание которых в литературе по языку **SQL**. Каждая инструкция выполняет определенные действия, например создать таблицу, добавить в таблицу новые данные.

Каждая инструкция начинается с команды, после команды идет предложение одно или несколько. Предложение описывает данные с которыми работает инструкция или содержит уточняющую информацию о действии выполняемой инструкцией. Каждое предложение начинается с ключевого слова. Одни предложения в инструкции являются обязательными, а другие нет. Структура и содержимое предложения могут меняться. Многие предложения могут меняться. Многие предложения содержат имена таблиц, столбцов. Выполнить следующие команды по созданию базы данных.

1. В своем каталоге создать новую папку для работы с новой базой данных. Имя папки выбрать самостоятельно.

2. Далее в командном окне набрать следующую команду

**SET DEFAULT. TO <путь>**

Например d:\Иванов \базы данных \вторая

После набора команды нажать enter. Будет установлен путь прямой для создаваемой информации.

3. Набрать команду

**CREATE DATABASE AVTO (Создание базы данных)**

4. **MODIFY DATABASE (модификация базы данных)**

5. **CREATE Автомобиль (создание таблицы Автомобиль)**

6. Создаем структуру таблицы:

владелец, модель, номер, год\_выпуска. Тип каждого поля выбрать самостоятельно.

7. Набрать команду либо: **BROWSE LAST** либо **APPEND**

8. Заполнить таблицу записями.

9. Сформировать запрос, выполнив следующую команду: **SELECT<список полей> from <имя таблицы > WHERE<условие отбора>**

Сделать выборку информации об автомобилях, выпущенных после 2000 года.



## Практическая работа № 5

### Разработка базы данных с текстовым файлом.

1. Создать базу данных, содержащую информацию о крупнейших фирмах мира:

Название фирмы	Страна	Производство	Оборот (млрд долл)	Количество сотрудников
Дженерал Моторс	США	автомобили	102	811000
Бритиш	Великобритания	нефтепродукты	45	126020
Тойота мотор	Япония	автомобили	42	84207
ИРИ	Италия	металла	41	422000
Тексако	США	нефтепродукты	100	54481
Эксон	США	нефтепродукты	76	146000

2. Создать форму, используя Мастер-форм, в качестве источника данных выбрать таблицу, содержащую сведения о фирмах. Форму сохранить в папке, где находится таблица, дать имя ФИРМА.

3. Создать текстовый файл, который будет содержать сведения об авторе базы данных. Выполнить следующие команды:

File- NEW-TEXT FILE - NEW file

В открывшемся окне набрать текст о разработчике базы данных. Далее сохранить файл file1.txt.

4. File- NEW - FORM - NEW file

Нанести на форму следующие кнопки Command Button:

1-ая фирма - свойство **caption** на **фирма**, активизировать объект двойным щелчком мыши и написать следующую команду  
do form фирма.scx

фирма.scx- это название файла, где хранится форма;

2-ая сведения - свойство **caption** на **сведения**, активизировать объект двойным щелчком мыши и написать следующую команду  
modify command file1.txt

3-я выход - свойство **caption** на **выход**, активизировать объект двойным щелчком мыши и написать следующую команду  
thisform.release

5. Выполнить форму, просмотреть результат.