Основные Возможности системы Visual FoxPro 7.0

СУБД FoxPro является в настоящее время одним из наиболее популярных средств реализации баз данных реляционного типа. Visual FoxPro 7.0 функционирует в среде Windows, полностью соответствует современным требованиям, предъявляемым к системам обработки информации, имеет событийно-управляемый объектно-ориентированный язык программирования и позволяет решать многие задачи визуально.

Характеристики таблиц и индексных файлов

* Максимальное количество записей в файле – 1 млрд.
* Максимальный размер файла таблицы – 2 Гбайт.
* Максимальное количество символов в записи – 65 500.
* Максимальное количество полей в записи – 255.
* Максимальное количество одновременно открытых таблиц – 255.
* Максимальное количество символов в поле таблицы – 254.
* Максимальное число байт в индексном ключе некомпактного индекса – 100.
* Максимальное число байт в индексном ключе компактного индекса – 240.
* Неограниченное число открытых индексных файлов таблицы.
* Неограниченное число открытых индексных файлов во всех рабочих областях.
* Неограниченное число устанавливаемых отношений.
* Неограниченная длина выражений, определяющих отношения.

Характеристики полей

* Максимальный размер символьного поля – 254.
* Максимальный размер числового поля – 20.
* Максимальная длина имени поля свободной таблицы – 10.
* Максимальная длина имени поля таблицы, содержащейся в базе данных – 128.
* Минимальное значение целого числа - -2 147 483 647.
* Максимальное значение целого числа - 2 147 483 647.
* Точность вычислений - до 16 разрядов.

Характеристики переменных и массивов

* Количество переменных по умолчанию – 1024.
* Максимальное количество переменных – 65 000.
* Максимальное количество массивов – 65 000.
* Максимальное количество элементов в массиве – 65 000.

Характеристики программных и процедурных файлов

* Неограниченное число строк в исходном программном файле.
* Максимальный размер скомпилированного программного модуля – 64 Кбайт.
* Неограниченное количество процедур в файле.
* Максимальное количество вложенных уровней оператора DO - 128.
* Максимальное количество вложенных уровней оператора READ – 5.
* Максимальное количество вложенных команд при структурном программировании – 384.
* Максимальное количество передаваемых параметров – 27.
* Максимальное количество транзакций – 5.

Характеристики Конструктора отчета

* Неограниченное количество объектов, обрабатываемых отчетом.
* Максимальная длина определения отчета – 20 дюймов.
* Максимальное количество уровней группирования – 128.
* Максимальная длина символьных переменных отчета – 255.

Другие возможности

* Неограниченное количество открытых окон различных типов.
* Максимальное количество открытых BROWSE-окон – 255.
* Максимальное количество символов в символьной строке или временной переменной –16 777 184.
* Максимальное количество символов в командной строке – 8192.
* Максимальное количество символов в замещаемой строке при использовании макросов – 8 192.
* Максимальное количество открытых файлов ограничивается только системой.
* Максимальное количество используемых клавиш, назначаемых в макросах – 1 024.
* Максимальное количество полей, обрабатываемых командой SELECT (SQL) – 255.

Основные Возможности Microsoft Office Access

Microsoft Office Access или просто Access это одна из самых популярных систем управления базами данных (СУБД) реляционного типа корпорации Microsoft. Данная СУБД имеет широкий спектр функций, таких как связанные запросы, сортировка по разным полям, связь с внешними таблицами и базами данных, а также встроенный язык Visual Basic for Applications (VBA). С помощью Access можно создавать приложения, работающие с базами данных. Одновременно Access может работать только с одной базой данных, но одна база данных Access может включать сотни таблиц, форм, запросов, отчётов, макросов и модулей, которые хранятся в одном файле с расширением accdb для Access 2007-2010 или с расширением mdb для предыдущих версий.

Основные общие характеристики Access 2010

* Максимальное значение размера файла базы данных Microsoft Access (accdb) - 2 ГБ за вычетом места, необходимого системным объектам
* Максимальное число объектов в базе данных - 32 768
* Максимальное число модулей (включая формы и отчеты, у которых свойство HasModule (наличие модуля) имеет значение True) - 1 000
* Максимальное число знаков в имени объекта - 64
* Максимальное число знаков в пароле - 20
* Максимальное число знаков в имени пользователя или имени группы - 20
* Максимальное число одновременно работающих пользователей - 255

Характеристики таблиц Access 2010

* Максимальное число знаков в имени таблицы - 64
* Максимальное число знаков в имени поля - 64
* Максимальное число полей в таблице - 255
* Максимальное число открытых таблиц - 2048 (фактическое число может быть меньше из-за внутренних таблиц, открытых в Access)
* Максимальный размер таблицы - 2 ГБ за вычетом места, необходимого системным объектам
* Максимальное число знаков в текстовом поле - 255
* Максимальное число знаков в поле MEMO - 65 535 при вводе данных через интерфейс пользователя;2 ГБ для хранения знаков при программном вводе данных
* Максимальный размер поля объекта OLE - 1 ГБ
* Максимальное число индексов в таблице - 32
* Максимальное число полей в индексе - 10
* Максимальное число знаков в сообщении об ошибке - 255
* Максимальное число знаков в условии на значение - 2 048
* Максимальное число знаков в описании таблицы или поля255
* Максимальное число знаков в записи (кроме полей MEMO и полей объектов OLE), если свойству Сжатие Юникод полей присвоено значение **Да** - 4 000
* Максимальное число знаков в значении свойства поля - 255

Основные характеристики запросов Access 2010

* Максимальное число установленных связей - 32 на одну таблицу за вычетом числа индексов, находящихся в таблице для полей или сочетаний полей, не участвующих в связях
* Максимальное число таблиц в запросе - 32
* Максимальное число объединений в запросе - 6
* Максимальное число полей в наборе записей - 255
* Максимальный размер набора записей - 1 ГБ
* Предел сортировки - 255 знаков в одном или нескольких полях
* Максимальное число уровней вложения запросов - 50
* Максимальное число знаков в ячейке на бланке запроса - 1 024
* Максимальное число знаков в параметре в запросе с параметрами - 255
* Максимальное число операторов AND в предложении WHERE или HAVING - 99
* Максимальное число знаков в инструкции SQL - приблизительно64 000

Некоторые из максимальных значений могут быть меньше, если запрос содержит поля подстановок, одновременно допускающих несколько значений.

Основные характеристики форм и отчетов Access 2010

* Максимальное число знаков в надписи - 2 048
* Максимальное число знаков в поле - 65 535
* Максимальная ширина формы или отчета - 22 дюйма (55,87 см)
* Максимальная высота раздела - 22 дюйма (55,87 см)
* Максимальная высота всех разделов вместе с заголовками разделов (в режиме конструктора) - 200 дюймов (508 см)
* Максимальное число уровней вложения форм или отчетов - 7
* Максимальное число полей или выражений, которые можно отсортировать или сгруппировать в отчете - 10
* Максимальное число заголовков и примечаний в отчете - 1 заголовок/примечание отчета; 1 заголовок/примечание страницы; 10 заголовков/примечаний групп
* Максимальное число печатных страниц в отчете - 65 536
* Максимальное число элементов управления и разделов, которые можно добавить за время существования формы или отчета – 754
* Максимальное число знаков в инструкции SQL, используемой в качестве свойства RecordSource (источник записей) или RowSource (источник строк) формы, отчета или элемента управления (как ACCDB, так и ADP) - 32 750

Основные характеристики макросов Access 2010

* Максимальное число макрокоманд в макросе - 999
* Максимальное число знаков в условии - 255
* Максимальное число знаков в комментарии – 255
* Максимальное число знаков в аргументе макрокоманды - 255

Основные общие характеристики проектов Access 2010:

* Максимальное число объектов в проекте Access (ADP) - 32 768
* Максимальное число модулей (включая формы и отчеты, у которых свойство HasModule (наличие модуля) имеет значение True) - 1 000
* Максимальное число знаков в имени объекта - 64
* Максимальное число столбцов в таблице - 250 (Microsoft SQL Server 6.5); 1024 (Microsoft SQL Server 7.0, 2000 и 2005)

Характеристики форм и отчетов проектов Access 2010:

* Максимальное число знаков в надписи2 048
* Максимальное число знаков в поле - 65 535
* Максимальная ширина формы или отчета - 22 дюйма (55,87 см)
* Максимальная Высота раздела - 22 дюйма (55,87 см)
* Максимальная Высота всех разделов вместе с заголовками разделов (в режиме конструктора) -
* 200 дюймов (508 см)
* Максимальное число уровней вложения форм или отчетов - 7
* Максимальное число полей или выражений, которые можно отсортировать или сгруппировать в отчете – 10
* Максимальное число заголовков и примечаний в отчете - 1 заголовок/примечание отчета; 1 заголовок/примечание страницы; 10 заголовков/примечаний групп
* Максимальное число печатных страниц в отчете - 65 536
* Максимальное число элементов управления и разделов, которые можно добавить за время существования формы или отчета – 754
* Максимальное число знаков в инструкции SQL, используемой в качестве свойства RecordSource (источник записей) или RowSource (источник строк) формы, отчета или элемента управления (как ACCDB, так и ADP) - 32 750

Характеристики макросов проектов Access 2010:

* Максимальное число макрокоманд в макросе - 999
* Максимальное число знаков в условии - 255
* Максимальное число знаков в комментарии - 255
* Максимальное число знаков в аргументе макрокоманды - 255

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**СОЗДАНИЕ ОТДЕЛЬНЫХ ТАБЛИЦ И СРЕДСТВА ИХ ВЕДЕНИЯ, ИНДЕКСИРОВАНИЕ ТАБЛИЦ, СОЗДАНИЕ БАЗЫ ДАННЫХ**

**Создание таблиц. Теоретические сведения.**

***Создание таблиц в Visual FoxPro 7.0***

В Visual FoxPro 7.0 можно создавать как таблицы, входящие в базу данных, так и отдельные таблицы, называемые свободными (free) таблицами. В этой лабораторной работе будут рассмотрены вопросы, связанные с созданием свободных таблиц.

Каждая таблица хранится в памяти на диске в виде файла с расширением .DBF (data base file), например, stud.dbf. Имена таблицы и файла с расширением .DBF совпадают, поэтому, при задании имени таблицы, необходимо учитывать ограничения на имя файла, которые определяются требованиями операционной системы.

Создание таблицы заключается в определении структуры таблицы и её заполнении. В свою очередь структура таблицы включает перечень имён полей таблицы (Name) с указанием типов (Type) и размерностей полей (Width – общая длина поля, Decimal – длина дробной части числа), признака пустого поля (Null), а также указание ключевых полей (Index).

Имена полей задаются как прописными, так и строчными буквами. Длина имени поля свободной таблицы не должна превышать 10 символов. Длина имени поля таблицы, входящей в базу данных, может находиться в диапазоне от 1 до 254 символов, при этом использование пробелов в имени не допустимо.

Основные типы полей Visual FoxPro 7.0.

|  |  |
| --- | --- |
| *Тип поля* | *Содержание* |
| Character (Символьный) | Любые символы. Прописные и строчные буквы обрабатываются по-разному. Максимальный размер символьного поля – 254 символа. |
| Numeric (Числовой) | Числа (целые или дробные), подлежащие математической обработке. Содержит числа от 0 до 9, а также знак (необязательно) и десятичную запятую (необязательно). Дробную часть числа вводить необязательно. Диапазон: от -.9999999999Е+19 до .9999999999Е+20. |
| Integer (Числовой, целочисленный) | Целые числа. Хранятся в таблицах как четырехбайтные двоичные значения и поэтому занимают меньше памяти, чем другие числовые типы данных, так как двоичные значения не требуют преобразования в ASCII. Диапазон от –2147483647 до 2147483646. |
| Float (Вещественный) | Числа, имеющие дробную часть (представляются в формате с плавающей точкой), определенные с обычной точностью. |
| Double (Вещественный с двойной точностью) | Числа, имеющие дробную часть (представляются в формате с плавающей точкой), определенные с двойной точностью. Если необходима большая точность, фиксированный объем памяти в таблице или значения с плавающей запятой, то лучше использовать этот тип вместо Numeric. Диапазон от +/-4.9406564584124Е-324 до +/-1.79769313486232Е+308. |
| Currency (Денежный) | Поле для представления денежных сумм. Используется для хранения чисел с четырьмя знаками после запятой. Если вы укажите более четырех десятичных цифр после десятичной запятой, то Visual FoxPro автоматически округлит их до четырех. Диапазон от –922337203685477,5808 до 922337203685477,5807. |
| Date (Дата) | Дата, определяющая день, месяц и год. Ввод в это поле контролируется автоматически. При вводе значения в поле по умолчанию принят формат даты – мм/дд/гг. Для изменения формата записи даты, например, на дд/мм/гг, необходимо ввести команду – Set Date Britisch. |
| DateTime (Дата и время) | Дата и время. Формат вводимого значения в поле базы данных по умолчанию имеет вид – мм/дд/гг чч:мм:сс. |
| Logical (Логический) | Логические данные. Используется для хранения информации, принимающей только два значения – «истина» (.T.) или «ложь» (.F.). Занимает в памяти 1 байт. |
| Memo (Текстовое поле произвольной длины) | Примечания, размер которых неограничен. Могут содержать любую символьную информацию. Фактически это указатель на блок данных в файле с расширением .FPT (файлы .FPT и .DBF одноименны). Доступ к файлу .FPT возможен только при открытом одноименном файле .DBF, поэтому при перемещении информации с одного носителя на другой необходимо копировать оба файла. Для ввода значения в поле нужно два раза щёлкнуть мышью по полю memo, после чего в открывшемся окне редактирования набрать нужный текст примечаний. Закрыв окно редактирования, значение поля примечаний будет сохранено, а в поле базы данных слово memo будет заменено на Memo, что является признаком его заполнения. |
| General (Двоичное поле произвольной длины) | Поле, в котором хранится и отображается OLE-объект (Object Linking and Embedding), созданный другими приложениями. OLE-объект может содержать графику, звук, видеоклип, документы Word for Windows, таблицы Microsoft Excel и т.п. Для ввода в поле базы данных объекта нужно выполнить двойной щелчок мыши на поле gen, после чего появится окно редактирования и изменится меню, затем выбрать пункт меню Edit-Insert Object и из перечня приложений, поддерживающих технологию OLE, выбрать нужное Вам (например рисунок Paintbrush). Закрыв окно приложения и окно редактирования поля, в базе данных поле gen будет заменено на Gen, что является признаком его заполнения. |
| Character binary | Используется для хранения двоичного текста длиной до 254 символов. Содержит любые символьные данные, которые вы хотите использовать, не изменяя кодовую страницу. |
| Memo binary | Поле примечаний для хранения двоичного текста произвольной длины. Содержит любые данные поля memo, которые вы хотите использовать, не изменяя кодовую страницу. |

В Visual FoxPro 7.0 существуют следующие способы создания таблицы:

1. Создание таблицы из текста программы.

Для этой цели предназначена команда SQL CREATE TABLE | DBF.

1. Создание таблицы с помощью командного окна.

Для создания таблицы необходимо определить, на каком диске, и в каком каталоге будет храниться создаваемая таблица.

Для получения информации об активном диске, а также каталоге, в командном окне надо выполнить команду вывода «?» и одну из следующих функций:

|  |  |
| --- | --- |
| CURDIR(<диск>) | Возвращает прописными буквами имя текущего каталога на активном или заданном диске |
| SYS(5) | Возвращает имя активного диска |
| SYS(2003) | Возвращает имя текущего каталога на диске |
| SYS(2004) | Возвращает полное имя системного каталога |

Установить текущий или активный каталог по умолчанию можно с помощью команды Set Default to <путь>, заданной в командном окне, или при помощи команды системного меню Tools/Options/File Locations/Default Dir…

Для создания таблицы в текущем каталоге, используется команда CREATE, при выполнении которой сначала открывается диалоговое окно Create, а затем открывается диалоговое окно Table Designer.

1. Создание таблицы с помощью конструктора.

Для создания таблицы необходимо выполнить команду системного меню File/New/Table, ввести имя таблицы в открывшемся диалоговом окне Create.

После присвоения имени таблицы открывается окно Table Designer, которое имеет три вкладки: Fields, Indexes, Table.

Для задания структуры таблицы используется вкладка Fields.

1. Создание таблицы с помощью мастера.

Для создания таблицы необходимо выполнить команду системного меню Tools/Wizard/Table.

В Visual FoxPro существует два режима просмотра таблицы: Edit и Browse.

По умолчанию после создания структуры таблицы устанавливается режим Edit. В этом режиме после ввода информации во все поля одной записи Visual FoxPro автоматически добавляет следующую запись.

В режиме Browse поля размещены в один ряд. Одна строка соответствует одной записи таблицы, а записи размещены одна под другой. При вводе данных в этом режиме после ввода одной записи пользователю необходимо самому добавлять новую пустую запись, используя для этого команду Table/Append New Record или комбинацию клавиш <Ctrl> + <Y>.

***Создание таблиц в Microsoft Access 2010***

Для создания файла базы данных необходимо выполнить команду меню Файл/Создать/Новая база данных. Далее именуем базу данных, определяем папку, в которой будет размещен этот файл и щелкаем кнопкой мыши на значке «Создать».

Для создания таблиц в Access существуют следующие способы:

1. Режим таблицы;
2. Конструктор;
3. Мастер таблиц;
4. Импорт таблиц;
5. Связь с таблицами.

***Создание таблицы через режим таблицы*** проходит путем непосредственного ввода данных в таблицу. При сохранении таблицы Microsoft Access проанализирует данные и автоматически присвоит каждому полю соответствующий тип данных и формат. Создание таблиц в режиме таблицы имеет ограниченное применение. Поля таблицы не имеют содержательных имен, так что почти всегда нужно будет редактировать структуру, чтобы переименовать поля. Кроме того, из примера Access не всегда может правильно определить тип данных. Таблицы, которые созданы в режиме таблицы, не могут включать объекты OLE и мемо-поля. Если есть желание иметь такие поля, вам нужно изменить структуру таблицы.

Для создания таблицы через режим таблицы необходимо:

1. Выполнить команду меню Создание/Таблица
2. Переименовать нужные столбцы
3. Ввести данные в таблицу
4. Сохранить данные (нажать кнопку «Сохранить» или Файл/Сохранить или знак «Закрыть таблицу»)
5. Создать первичный ключ.

В случае необходимости дополнительные столбцы можно добавлять в любое время.

***Создание таблицы в режиме конструктора*** позволяет добавлять поля, настраивать отображение полей и обработку в них данных, а затем создавать первичный ключ.

Для создания таблицы в режиме конструктора необходимо:

1. Выполнить команду меню Создание/Конструктор таблиц
2. Определить все нужные поля в таблице. Для этого:

* Щелкните ячейку в столбце **Имя поля** и введите имя поля.

*При задании имен полей в таблицах необходимо придерживаться ряда правил:*

1. Имена полей в таблице не должны повторяться, т. е. должны быть уникальными.
2. Имена полей могут содержать не более 64 символов, включая пробелы.
3. Желательно избегать употребления имен полей, совпадающих с именами встроенных функций или свойств Microsoft Access (например, Name — имя).
4. Имя поля не должно начинаться с пробела или управляющего символа (коды ASCII 00-31).
5. Имена полей могут содержать любые символы, включая буквы, цифры, пробелы, специальные символы, за исключением точки (.), восклицательного знака (!), апострофа (') и квадратных скобок ([), (]).

Рекомендуется не использовать очень длинные имена, т. к. они плохо запоминаются, и не включать в имена большое количество пробелов, особенно если с именами предполагается работать с помощью процедур Visual Basic for Application. Чтобы следовать этой рекомендации и при этом получать понятные имена, можно либо использовать вместо пробелов между словами знак подчеркивания (\_), либо писать все слова слитно, но каждое начинать с заглавной буквы.

Эти же правила действительны и для имен таблиц, а также других объектов Microsoft Access.

* В столбце **Тип данных** можно выбрать из раскрывающегося списка тип данных.

В Microsoft Access имеются следующие основные типы данных:

* + - **Текстовый** (Text) — символьные или числовые данные, не требующие вычислений. Поле данного типа может содержать до 255 символов. Размер текстового поля задается с помощью свойства Размер поля (FieldSize), в котором указывается максимальное количество символов, которые могут быть введены в данное поле. При необходимости ввести строку большего размера, чем указано в данном свойстве, надо значение свойства изменить, т. к. место под дополнительные символы не резервируется. Независимо от длины текстового поля оно хранится в базе данных Access в записи переменной длины. Access не занимает память под неиспользуемую часть текстового поля в записи, что уменьшает общий размер базы данных.
    - **Поле MEMO** (MEMO) — поле MEMO предназначено для ввода текстовой информации, по объему превышающей 255 символов. Такое поле может содержать до 65 535 символов. Этот тип данных отличается от типа **Текстовый** (Text) тем, что в таблице хранятся не сами данные, а ссылки на блоки данных, хранящиеся отдельно. За счет этого ускоряется обработка таблиц (сортировка, поиск и т. п.). Поле типа MEMO не может быть ключевым или проиндексированным. Не используйте поле типа MEMO, если нужно будет сортировать записи таблицы по данному полю — это невозможно. Если 255 символов не достаточно для хранения всех данных, придется разбить его на два или три текстовых поля.

Ввод данных в это поле можно выполнить непосредственно в поле таблицы, либо через область ввода, вызываемую нажатием комбинации клавиш <Shift>+<F2>.

* + - **Числовой** (Number) — числовой тип применяется для хранения числовых данных, используемых в математических расчетах. Имеет много подтипов. От выбора подтипа (размера) данных числового типа зависит точность вычислений. Для установки подтипа числовых данных служит свойство **Размер поля** (FieldSize). Данные этого типа могут содержаться в 1, 2, 4, 8 или 16 байтах. Обычно по умолчанию используется подтип **Длинное целое** (Longlnteger), который занимает 4 байта и представляет собой число в пределах от -2 147 483 648 до +2 147 483 647. Но, кроме этого типа, можно указать **Байт** (Byte) — 1 байт, **Целое** (Integer) — 2 байта, **Одинарное с плавающей точкой** (Single) — 4 байта, **Двойное с плавающей точкой** (Double) — 8 байтов, **Десятичное** (Decimal) — 12 байтов, **Код репликации** (Replication ID) — 16 байтов.
    - **Дата/Время** (Date/Time) — тип для представления даты и времени. Позволяет вводить даты с 100 по 9999 год. Размер поля — 8 байтов. Даты и время хранятся в специальном фиксированном числовом формате. Дата является целой частью значения поля типа Дата/Время, а время — его дробной частью. Access предоставляет большой выбор форматов отображения даты и времени.
    - **Денежный** (Currency) — тип данных, предназначенный для хранения данных, точность представления которых колеблется от 1 до 4 десятичных знаков. Целая часть данного типа может содержать до 15 десятичных знаков.
    - **Счетчик** (AutoNumber) — поле содержит 4-байтный уникальный номер, определяемый Microsoft Access автоматически для каждой новой записи либо случайным образом, либо путем увеличения предыдущего значения на 1. Значения полей типа счетчика обновлять нельзя. Максимальное число записей в таблице с полем счетчика не должно превышать двух миллиардов.
    - **Логический** (Yes/No) — логическое поле, которое может содержать только два значения, интерпретируемых как Да/Нет, Истина/Ложь, Включено/Выключено. Поля логического типа не могут быть ключевыми, но их можно индексировать. Access использует величину -1 для представления значения **Истина** и величину 0 — для значения **Ложь.**
    - **Поле объекта OLE** (OLE object) — содержит ссылку на OLE-объект (лист Microsoft Excel, документ Microsoft Word, звук, рисунок и т. п.). Объем объекта ограничивается имеющимся в наличии дисковым пространством. Тип объекта OLE не указывается в свойствах поля объекта OLE. Он зависит от приложения, использованного для создания объектов OLE, которые хранятся в этом поле. Упаковщик объектов позволяет внедрять файлы, созданные приложениями, которые не являются серверами объектов OLE. Например, он позволяет внедрить текстовые файлы в качестве объектов в таблицы Access 2002. В поле объекта OLE могут храниться произвольные данные, в том числе и данные нескольких типов. Это позволяет обойти основное ограничение реляционных баз данных, которое требует, чтобы в каждом поле хранились данные только одного типа.
    - **Гиперссылка** (Hyperlink) — дает возможность хранить в поле ссылку, с помощью которой можно ссылаться на произвольный фрагмент данных внутри файла или Web-страницы на том же компьютере, в интранет или в Интернет. Гиперссылка состоит из четырех частей: отображаемый текст, адрес (путь к файлу или странице), дополнительный адрес (положение внутри файла или страницы) и текст всплывающей подсказки. Каждая часть гиперссылки может содержать до 2048 символов. Поле типа **Гиперссылка** не может быть ключевым или индексированным.

В поле типа можно также выбрать значение **Мастер подстановок** (Lookup Wizard), который запускает Мастера подстановок (Lookup Wizard), создающего поле подстановок. *Поле подстановок* позволяет выбирать значение поля из списка, содержащего набор постоянных значений или значений из другой таблицы.

Как только выбран тип данных для поля, так на нижней панели окна отображается список свойств этого поля. Данный список зависит от типа поля. Для того чтобы просмотреть или изменить свойства конкретного поля таблицы необходимо:

1. Выделить нужное поле таблицы (то есть установить курсор в строку с именем поля) в окне Конструктора.
2. Раскрыть вкладку Общие (General) для просмотра общих свойств поля или вкладку Подстановка (Lookup) для просмотра параметров подстановки.

* В столбце **Описание** введите описание данных, которые будет содержать это поле. Текст описания будет выводиться в строке состояния при добавлении данных в поле, а также будет включен в описание объекта таблицы. Вводить описание не обязательно.

1. Создать первичный ключ.

Чтобы установить для таблицы первичный ключ и завершить ее создание в режиме конструктора, необходимо выполнить следующие действия:

– в режиме конструктора выделить поле, которое будет играть роль первичного ключа;

– кликнуть по кнопке Ключевое поле панели инструментов Конструктора таблиц или выбрать команду главного меню Правка – Ключевое поле (слева возле имени выделенного поля появится символ ключа);

– после указания ключевого поля таблицу необходимо сохранить, для чего необходимо кликнуть по кнопке Сохранить панели инструментов конструктор таблиц и в открывшемся окне ввести имя таблицы и кликнуть по кнопке Ок.

Если первичный ключ не определен, то при выходе из режима конструктора появится соответствующее предупреждение, и программа Access предложит Вам создать ключевое поле до закрытия таблицы.

***Основные правила, которые приняты для ключей в Access:***

* для удобства ключевое поле обычно указывается в структуре таблицы первым;
* если для таблицы определен первичный ключ, программа Access автоматически блокирует ввод в это поле повторяющихся значений или значения Null (пусто);
* Access автоматически сортирует записи таблицы по первичному ключу;
* поле первичного ключа является индексом, с помощью которого ускоряется сортировка и поиск записей.

1. Сохранить таблицу (нажать кнопку «Сохранить» или Файл/Сохранить или знак «Закрыть таблицу»).

***Для создания таблицы с помощью мастера необходимо:***

* В 1-ом окне мастера можно выбрать категорию и образец таблицы, затем нужные образцы полей. Допускается выбор полей из нескольких таблиц.
* Во 2-ом окне мастера можно выбрать имя для таблицы, а также способ определения ключа. Все сведения, необходимые для создания таблицы, указаны.
* В 3-ем окне мастера выбирают дальнейшие действия после создания таблицы.
* Нажимают кнопку Готово.

Если в дальнейшем потребуется изменить или расширить созданную с помощью мастера таблицу, это можно сделать в режиме конструктора.

**Задание №1**

***В Visual FoxPro 7.0***

1. Укажите собственную директорию: Tools/Options/File Locations/Default Dir... или задайте в командном окне команду: Set Default to <имя каталога>.
2. Создайте таблицу, содержащую сведения о студентах с именем Stud.dbf (File-New-Table), имеющую следующие поля:

* номер зачётной книжки (Name-Num, Type-Numeric, Width-6)
* фамилия и инициалы (Name-Fio, Type-Character, With-25)
* дата рождения (Name- DR, Type-Date)
* номер группы (Name-N\_gr, Type-Character, With-6)
* год поступления в институт (Name-God, Type - Integer),
* вид обучения - бюджет или коммерческое (Name-Plata, Type- Logical)
* количество баллов при поступлении в институт (Name-Sum\_b, Type- Integer). Может быть неизвестно. При интерактивном редактировании данных значение NULL вводится нажатием клавиш CTRL+0. Для определения отображения пустых значений при работе с окнами Browse, командами DISPLAY и LIST используется команда:

SET NULLDISPLAY TO [текст, отображающий пустое значение]

По умолчанию пустые значения отображаются словом NULL.

* размер оплаты при поступлении на коммерческой основе (Name-Pay,Type-Currency)
* портрет (Name-Photo, Type-General)
* адрес (Name-Adr, Type-Memo).

1. Заполнить 5-10 записями созданную таблицу.
2. Просмотреть записи в таблице, используя режим формуляра (View-Edit) и табличный режим (View-Browse).
3. Добавить, отметить на удаление и удалить записи таблицы, используя соответствующие пункты меню Table для открытой таблицы, находящейся в режиме просмотра.

***В Microsoft Access 2010***

Создать одноименную таблицу, содержащую сведения о студентах и подобрав соответствующие типы полей.

**Вопросы.**

1. Какие действия необходимо выполнить при создании структуры таблицы?
2. Какие основные типы файлов имеются в Visual FoxPro 7.0?
3. Какие типы полей используются в Visual FoxPro 7.0 и Microsoft Access?
4. Что Вы понимаете под полями с NULL значениями?
5. Как осуществляется ввод данных в Memo – поле в Visual FoxPro 7.0 и Microsoft Access?
6. Когда выбирается режим таблицы, и в каких случаях производится переключение на режим формуляра в Visual FoxPro 7.0?
7. Какие особенности демонстрирует FoxPro при удалении записей, и в чем преимущество существующего механизма удаления?
8. Какая команда выполняет физическое удаление отмеченных записей Visual FoxPro 7.0?

**Индексирование таблиц. Теоретические сведения.**

С целью ускорения поиска и сортировки данных в любой СУБД используются индексы. Индекс является средством, которое обеспечивает быстрый доступ к данным в таблице на основе значений одного или нескольких столбцов. Индекс представляет собой упорядоченный список значений и ссылок на те записи, в которых хранятся эти значения. Чтобы найти нужные записи, СУБД сначала ищет требуемое значение в индексе, а затем по ссылкам быстро отбирает соответствующие записи. Индексы бывают двух типов: простые и составные. Простые индексы представляют собой индексы, созданные по одному столбцу. Индекс, построенный по нескольким столбцам, называется составным. Однако применение индексов приносит не только преимущества, но и недостатки. Главным среди них является тот, что при добавлении и удалении записей или при обновлении значений в таблице требуется обновлять индекс, что при большом количестве индексов в таблице может замедлять работу. Поэтому индексы обычно рекомендуется создавать только для тех столбцов таблицы, по которым наиболее часто выполняется поиск записей.

***Индексирование таблиц в Visual FoxPro 7.0.***

Для использования алгоритмов поиска, обеспечивающих большую эффективность, а также для связывания нескольких таблиц по общему признаку, выполняется их индексирование по одному полю или комбинации полей (кроме полей типа Memo и General).

В Visual FoxPro можно создавать файлы индексов двух типов:

• Файлы составных индексов .CDX, которые состоят из множества элементов индекса, называемых тегами.

• Файлы индексов .IDX, каждый из которых содержит только один элемент индекса.

Вы также можете создать файл структурного составного индекса, который автоматически открывается вместе с таблицей.

Каждый индекс содержится в отдельном файле, в котором для каждого индекса существует уникальная ссылка, указывающая на местонахождение записи, соответствующей индексу, или является одним из тегов мультииндексного файла.

В Visual FoxPro поддерживаются следующие типы индексов:

* Primary (первичный) - используется для создания первичных ключей. Этот индекс может быть только один и не может содержать повторяющихся значений. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений.
* Candidate (кандидат) так же, как и первичный индекс, не допускает повторяющихся значений, и обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа. Любая таблица может иметь более одного индекса этого типа.
* Regular (обычный) может содержать повторяющиеся значения, при этом создается индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Используется для установления связи «многие к одному» в той таблице, где находятся «многие».
* Unique (уникальный) является обычным индексом, который показывает только одну запись для каждого значения индексного выражения. Уникальный индекс, в отличие от первичного индекса и индекса типа «кандидат», не требует уникальности значений, то есть можно вводить записи с повторяющимися значениями индексного поля. Разница между обычным и уникальным индексом заключается в том, что если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с одинаковым значением индексного выражения. Таблица может иметь несколько уникальных индексов.

Если таблица проиндексирована, то такие команды как DISPLAY, LIST, BROWSE, REPLACE, SKIP, GO перемещают указатель записей в соответствии с индексом, а не физическим порядком записей в таблице.

Индексирование и создание индексного файла с расширением .IDX для открытой таблицы выполняется командой

INDEX ON < врж > TO <имя индексного файла> [FOR < вржL >][DESCENDING] [UNIQUE] [ADDITIVE] [COMPACT]

В качестве <врж> чаще всего используют просто имя поля, по которому упорядочивают таблицы, но индекс может быть и сложным. При этом порядок просмотра записей в таблице определяется полем, имя которого указано в выражении первым.

В команде индексирования опция DESCENDING используется для упорядочивания записей по убыванию значений <врж> (по умолчанию – по возрастанию).

Опция UNIQUE позволяет игнорировать записи с повторяющимся значением <врж>.

Опция ADDITIVE используется для того, чтобы при создании индекса не закрывались уже открытые индексные файлы.

Опция COMPACT сохраняет индексируемые данные в строках фактической, а не фиксированной длины, а повторяющиеся элементы при этом записываются в одном экземпляре, что позволяет сократить размер таблицы в 5-6 раз.

Индексные файлы с расширением .IDX, относящиеся к одной открытой таблице могут быть объединены как теги в один мультииндексный файл с расширением .CDX, который синхронизирует работу сразу нескольких индексов. Мультииндексный файл создается командой

COPY INDEXES < перечень файлов .IDX > ALL |

[TO < имя неструктурированного файла .СDX >]

Если опция TO < > не используется, то по умолчанию создается структурированный мультииндексный файл, который открывается всякий раз одновременно с таблицей и получает ее имя, но имеет расширение .СDX.

Неструктурированный .CDX - файл не открывается автоматически при открытии таблицы, его нужно открывать специально, если нужен определенный способ упорядочения записей в базе данных.

Для создания нового тега или дополнения файла с расширением .СDX новым тегом используют команду

INDEX ON < врж > TO TAG < имя тега >

[OF < имя неструктурированного файла .СDX - файла > ]

[FOR < вржL >] [DESCENDING] [UNIQUE] [ADDITIVE]

Индексные файлы с расширением .IDX при активной таблице могут быть открыты командой

SET INDEX TO < список индексных файлов > [ADDITIVE]

Управляющий индекс можно переназначить командой

SET ORDER TO < имя .IDX – файла > |

< номер .IDX - файла > | [TAG] < имя тега >

[DESCENDING] [ADDITIVE]

Номер .IDX - файла указывает один из файлов индексов в соответствии с порядком, в котором они перечислены в команде USE или SET INDEX. Сначала нумеруются открытые файлы .IDX в порядке, в котором они указаны в команде USE или SET INDEX. Затем нумеруются теги в файле структурного индекса .CDX (если таковой существует) в том же порядке, в каком они создавались. Наконец, нумеруются теги во всех открытых файлах независимых индексов .CDX в том же порядке, в каком они создавались.

Если индексный файл уже создан, его можно открыть одновременно с таблицей командой

USE < имя таблицы > IN < номер области >

INDEX < список .idx, .cdx – файлов > [AGAIN] [NOUPDATE]

Подробный формат команд и их описание можно посмотреть в Help.

***Индексирование таблиц в Microsoft Access 2010***

В Microsoft Access 2010 создать индексы, как и ключи, можно по одному или нескольким полям. Составные индексы позволяют при отборе данных группировать записи, в которых первые поля могут иметь одинаковые значения. Индексировать поля требуется для выполнения частых поисков, сортировок или объединений с полями из других таблиц в запросах. Ключевые поля таблицы индексируются автоматически. Индексировать можно любые поля, кроме МЕМО-полей, полей типа Гиперссылка и объектов OLE. Если предполагается, что будет часто выполняться сортировка или поиск одновременно по двум и более полям, можно создать составной индекс. При сортировке таблицы по составному индексу сначала осуществляется сортировка по первому полю, определенному для данного индекса. Если в первом поле содержатся записи с повторяющимися значениями, то сортировка осуществляется по второму полю.

Для создания индекса по одному поля необходимо:

1. В режиме конструктора в панели структуры таблицы (верхняя часть окна) выбрать поле, для которого требуется создать индекс.

2. В панели свойств (нижняя часть окна) для свойства Индексированное поле установить значение "Да (Допускаются совпадения)" или "Да (Совпадения не допускаются)". Убедиться, что в данном поле совпадающих записей нет, можно выбрав значение "Да (Совпадения не допускаются)".

Для создания составного индекса необходимо:

1. В режиме конструктора на панели инструментов нажать кнопку Индексы.

2. В первой пустой строке поля Индекс ввести имя индекса. Для индекса можно использовать либо имя одного из индексируемых полей, либо другое подходящее имя.

3. В поле Имя поля нажать стрелку и выбрать в списке первое поле, для которого требуется создать индекс.

4. В следующей строке поля Имя поля указать второе индексируемое поле (для данной строки поле Индекс следует оставить пустым). Повторить эти действия для всех полей, которые необходимо включить в индекс. В составном индексе может быть до 10 полей.

В Microsoft Access индексы хранятся в том же файле базы данных, что и таблицы и другие объекты Access.

**Задание №2**

***В Visual FoxPro 7.0***

1. Проиндексируйте таблицу Stud.dbf по номеру группы (поле N\_gr), создав индексный файл с расширением .idx.
2. Просмотрите таблицу и выполните все возможные варианты перехода к указанной записи, обратив внимание на то, как изменится перемещение указателя записей.
3. Проиндексируйте таблицу Stud.dbf по фамилии стедента (поле Fio), создав индексный файл с расширением .idx.
4. Выполните индексирование таблицы Stud.dbf сначала по номеру группы (поле N\_gr), а внутри группы по фамилии студента (поле).
5. Выполните установку индексов с использованием Table Designer. Создайте индекс типа «кандидат» по номеру зачетки (поле Num) и обычный индекс по номеру группы (поле N\_gr). Обратите внимание, что все теги, созданные в Table Designer, автоматически вносятся в структурированный, т.е. одноименный с таблицей мультииндексный файл с расширением .CDX, который используется только для ускорения выполнения таких команд, как SEEK, REPORT и т.д., а не для визуального упорядочения или выборки данных.

***В Microsoft Access 2010***

Создать аналогичные индексы для таблицы, содержащей сведения о студентах.

**Вопросы.**

1. Какие типы файлов индексов существуют в Visual FoxPro ?
2. В чём различие между простым индексным и мультииндексным файлами?
3. Какие типы индексов существуют в Visual FoxPro и Microsoft Access?
4. Как осуществляется перемещение указателя записей в проиндексированной таблице?
5. Какой порядок нумерации индексов в Visual FoxPro?
6. Что такое структурированный мультииндексный файл в Visual FoxPro?

**Создание баз данных как совокупности взаимосвязанных таблиц. Теоретические сведения.**

***Создание базы данных в Visual FoxPro 7.0.***

Реляционная база данных – это совокупность взаимосвязанных таблиц, описывающих некоторую предметную область. Под предметной областью понимается часть реального мира, рассматриваемого в данной системе.

Может ли база данных состоять из одной таблицы или включать кроме совокупности взаимосвязанных таблиц отдельные таблицы? Да может, если они относятся к одной предметной области и удовлетворяют требованиям нормализации отношений.

При создании базы данных важным понятием является понятие рабочей области. Рабочая область - это область памяти с одним файлом базы данных и несколькими индексными или мультииндексными файлами.

Visual FoxPro 7.0 имеет 32767 рабочих областей. При запуске Visual FoxPro 7.0 по умолчанию активизируется рабочая область 1 (или a).

В указанной рабочей области таблица открывается командой

USE <имя таблицы> IN <номер рабочей области> ALIAS <псевдоним>

Рабочие области и файлы таблиц базы данных могут идентифицироваться псевдонимами. Псевдонимом области по умолчанию является само имя находящегося в ней файла базы данных. В качестве псевдонима можно указать и любое другое слово в команде USE. Область, в которой мы находимся в данный момент, называется активной рабочей областью, и в ней можно работать с находящейся в ней таблицей, выполняя все допустимые команды системы. Одновременно даже в одной команде можно иметь доступ к полям других таблиц базы данных. В этом случае имя поля из неактивной области – составное. Собственно имени поля предшествует имя рабочей области или псевдоним, разделенные знаками «-», или (что более удобно) точкой:

<рабочая область | псевдоним> - <имя поля>

<рабочая область | псевдоним> .<имя поля>

При работе с таблицами, расположенными в нескольких рабочих областях и не связанными между собой, указатели по записям работают независимо друг от друга.

Если открыто несколько таблиц без указания рабочей области, то только последняя будет активной, поскольку будет открыта в рабочей области 1.

Активная рабочая область выбирается командой:

SELECT <номер | имя рабочей области>

| <имя таблицы> | <псевдоним>

Если необходимо открыть несколько таблиц в разных рабочих областях, удобно использовать команду SELECT 0, которая автоматически выбирает первую свободную рабочую область.

В Visual FoxPro 7.0 базой данных называется контейнер – таблица с расширением .dbc, которая содержит информацию о включенных в нее таблицах, индексах, отношениях между таблицами, представлениях, процедурах и функциях.

База данных позволяет создавать и сохранять отношения, установленные между таблицами, и при необходимости отображать их.

Каждая таблица может принадлежать только одной базе данных.

База данных может быть создана командой

CREATE DATABASE < имя >

или командой File/New/DataBase/New меню Visual FoxPro 7.0

Добавить таблицу к базе данных можно любым из трех следующих способов:

* Выполнить команду меню DataBase/New или DataBase/Add Table;
* Щелкнуть на кнопке New или Add Table панели инструментов DataBase Designer;
* Выбрать команду New или Add Table в контекстном меню области базы данных.

В Visual FoxPro между таблицами, входящими в базу данных, могут быть установлены связи типа один к одному и один ко многим. Связи типа многие ко многим преобразуются к связям типа один ко многим и многие к одному на более ранних этапах проектирования баз данных. Для установления постоянной связи между двумя таблицами необходимо предварительно проиндексировать каждую из них, причем поле, по которому будет установлена связь, должно быть одного типа и одной размерности. При установлении связи между двумя таблицами одна из них является главной, а другая подчиненной. Главную таблицу называют родительской (это управляющая таблица, то есть Вы манипулируете ею непосредственно), а подчиненную - дочерней (или таблицей-потомком, которой управляют через связь с родителем).

Для установления связи 1:1 типы индексов в каждой из связываемых таблиц должны быть или типа Candidate или типа Primary. Для установления связи 1:N тип индекса родительской таблицы (соответствующей односвязной сущности) должен быть или Primary или Candidate, а дочерней (соответствующей N - связной сущности) или Regular или Unique.

Для создания отношений между таблицами, включенными в базу данных, необходимо подвести мышь к индексному полю родительской таблицы, нажать на это поле и, не отпуская перенести на соответствующее индексное поле дочерней таблицы, по которому устанавливается связь. Таким образом устанавливаются необходимые связи между всеми таблицами, входящими в базу данных.

Необходимо отметить, что отношения, созданные в Конструкторе баз данных сохраняются и после окончания работы с системой, требуют индексации всех таблиц, и не управляют синхронным перемещением курсора, а могут быть использованы для создания представлений и запросов.

Для установления временных связей между таблицами, то есть связей существующих во время одного сеанса работы с FoxPro, необходимо также проиндексировать связываемые таблицы по общим полям, затем выполнить последовательность действий – Window/Data Session и в окне Data Session открыть нужные таблицы. Для установления связи 1:1 подвести курсор к родительской таблице, нажать кнопку Rеlations и подвести курсор к дочерней таблице, после выбора, которой в правой части окна будет показана установленная связь. Для установления временной связи 1:N необходимо установить связь 1:1, затем подвести курсор к родительской таблице и нажать на кнопку «1 To Many», после чего в правой части окна будет показана связь типа 1:N между соответствующими таблицами. При просмотре связанных таблиц переход по записям осуществляется синхронно, с учётом установленных связей.

***Создание базы данных в Microsoft Access 2010***

Механизм описания логических связей между таблицами в Access реализован в виде объекта, называемого Схемой данных. Перейти к ее созданию можно из панели инструментов Работа с базами данных/Схема данных, доступной из главного окна.

Для добавления таблицы в схему базы данных необходимо:

1. Щелкнуть правой кнопкой мыши на свободном пространстве схемы данных и из контекстного меню выбрать команду Добавить таблицу.
2. В диалоговом окне **Добавление таблицы** раскрыть вкладку Таблицы и выбрать одну или несколько таблиц из списка таблиц, и нажать кнопку Добавить.

Для установления связей между таблицами необходимо:

1. Перетащите поле (как правило, поле первичного ключа) из одной таблицы на общее поле (поле внешнего ключа) в другой таблице. Чтобы перетащить сразу несколько полей, нажмите клавишу CTRL и, удерживая ее, выберите каждое поле.
2. В открывшемся диалоговом окне **Изменение связей** при необходимости задайте условия обеспечения целостности данных и объединения и нажмите кнопку OK.

Access проведет линию связи между двумя таблицами. Если был установлен флажок Обеспечение целостности данных, линия будет толще на каждом конце. Кроме того, если был установлен флажок **Обеспечение целостности данных**, над широкой частью на одном конце линии связи будет отображено число 1, а над широкой частью на другом конце линии — символ бесконечности (∞).

Для создания отношения «один-к-одному» оба общих поля (как правило, поля первичного ключа и внешнего ключа) должны иметь уникальный индекс. Это означает, что свойства Индексированно этих полей должны иметь значения Да (Совпадения не допускаются). Если оба поля имеют уникальный индекс, Access создаст отношение «один-к-одному».

Для создания отношения «один ко многим» поле на одной стороне отношения (как правило, поле первичного ключа) должно иметь уникальный индекс. Это означает, что свойство Индексировано этого поля должны иметь значения Да (Совпадения не допускаются). Поле на стороне «многие» не должно иметь уникального индекса. У этого поля может быть индекс, но он должен допускать совпадения. Это означает, что свойство Индексировано этого поля должно иметь значение Нет либо Да (Допускаются совпадения). Если одно поле имеет уникальный индекс, а другое — нет, Access создаст отношение «один ко многим».

Для изменения связи между таблицами следует выбрать ее на вкладке «Схема данных», а затем изменить эту связь.

**Задание №3**

***В Visual FoxPro 7.0***

1. Создать базу данных Base1.DBC, хранящую информацию о студентах, группах, изучаемых дисциплинах и паспортные данные студентов. Для этого добавьте созданную ранее таблицу Stud.dbf и создайте следующие новые таблицы:

* Группа (номер группы, специальность)
* Дисциплина (код дисциплины, название дисциплины)
* Группа – Дисциплина (номер группы, код дисциплины, № семестра кол-во часов, вид аттестации)
* Паспортные данные (номер зачетки, серия паспорта, номер паспорта, кем выдан, дата выдачи)

Установите связи между таблицами (Группа - Студент - 1:N, Студент - Паспортные данные – 1:1, Группа – Дисциплина N:M).

1. Установить временные связи между таблицами, после чего просмотреть через Browse, как установленные связи отображается при движении по записям в таблицах.

***В Microsoft Access 2010***

Создать аналогичную базу данных средствами Microsoft Access.

**Вопросы.**

1. Что такое рабочая область в Visual FoxPro?
2. Как создать базу данных, установив постоянные связи между таблицами в Visual FoxPro и Microsoft Access?
3. Какие типы связей реализуются в Visual FoxPro и Microsoft Access?
4. В чём отличие связей один к одному и один ко многим?
5. Как реализовать связи многие ко многим?
6. Как организовать временную связь между файлами базы данных в Visual FoxPro?