

Информатика

Лекция 4

Visual Basic.

Работа с базами данных

Реляционная база данных

- Данные в РБД представлены в виде таблиц (table), каждая таблица имеет произвольное количество строк-записей (record) и фиксированный состав столбцов (column), т.е. таблица состоит из упорядоченных элементарных ячеек (cell) с информацией.
- Столбец имеет тип (целый, текстовый и т.д.), определяющий, какого рода информацию могут содержать принадлежащие ему ячейки.
- Информация в таблице обрабатывается построчно, в каждый момент времени в непустой таблице имеется одна и только одна запись, к элементам (употребляется также термин к полям) которой может производиться доступ. Эта запись называется текущей (current record).

Реляционная база данных (2)

- Для упорядочивания записей в таблице с целью установления последовательности обработки создаются так называемые индексы. В VB индексы используются так же, как указатели (указатель – это index по-английски) в книгах: чтобы найти данные, осуществляется их поиск в индексе. Можно создать индексы, основанные на одном или нескольких полях таблицы.
- Если предполагается, что сортировка или поиск двух и более полей одновременно будет часто выполняться, можно создать составной индекс. Например, если для одного и того же запроса часто устанавливается критерий для полей “Имя” и “Фамилия”, то для этих двух полей имеет смысл 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, определенному для данного индекса, и так далее.

Оператор With

With объект

Операторы

End With

- Конструкция With – End With позволяет выполнить последовательность операторов над указанным объектом, не повторяя задание имени объекта. Например, если имеется несколько свойств, которые необходимо изменить для одиночного объекта, то удобнее поместить инструкции присвоения свойств внутрь управляющей структуры With, указав ссылку на объект один раз, вместо того чтобы ссылаться на объект при каждом присвоении его свойств.

With Label1

- Height = 2000
- Width = 2000
- Caption = "Объект MyLabel"

End With

- Внимание! Нельзя выполнять переходы внутрь или из блоков With. Если не выполнены инструкции With или End With, возможно возникновение ошибок или непредсказуемое поведение объектов.

Visual Data Manager

- Программа-утилита Visual Data Manager вызывается с помощью команды из меню Add-Ins. Интерфейс программы достаточно прост и требует лишь некоторые пояснения. Меню File рассматриваемой утилиты предоставляет стандартные возможности по работе с файлами. Для открытия существующей БД следует выполнить Open – Database и в появившемся дополнительном меню выбрать тип открываемой базы данных. Выберите Microsoft Access. При создании новой БД (пункт меню New) дополнительно укажите версию 7.0.
- После открытия базы данных можно создать новую таблицу, расположив курсор мыши на свободном месте в окне Database Window и нажав правую кнопку мыши.
- Окно “Table structure” (структура таблицы) позволяет работать с полями таблицы, в том числе добавлять в таблицу поля. Каждое поле таблицы имеет собственный тип. Эти типы в целом совпадают по наименованиям с типами данных VB, но есть и некоторые отличия. Например, вместо типа String (VB) имеется тип Text, введен тип Memo для текстовых данных произвольной (очень большой) длины.
- После ввода всей необходимой информации о полях таблицы следует присвоить ей имя и создать (выполнить команду Create Table).
- Для работы с существующей таблицей следует нажать правую кнопку мыши, расположив курсор на ее имени. В появившемся меню следует выбрать одну из команд: Open (открыть, т.е. рассмотреть записи таблицы), Design (модифицировать структуру таблицы), Rename (переименовать), Delete... . Для работы со структурой таблиц (т.е. для изменения состава полей таблицы, для редактирования их параметров) выберите Design, для просмотра и редактирования содержимого – Open.

Элемент интерфейса Data

Данный элемент интерфейса используется для установления связи между программой и базой данных. Data позволяет выводить записи и манипулировать ими, передвигаться от записи к записи. Вы можете выполнять большинство операций по доступу к данным без написания программного кода. Для организации доступа к данным VB создает из физической базы данных новый объект Recordset – набор записей. В простейшем случае, при использовании "родных" баз данных, набор записей содержит все записи соответствующей таблицы. После размещения на экранной форме элемента Data следует установить два обязательных свойства, которые и определяют источник данных:

- Свойство DatabaseName - имя базы данных.
- Свойство RecordSource – имя таблицы базы данных.

Кроме того, связью с физической базой данных управляет свойство Connect, определяющее тип базы данных, к которой производится подключение. По умолчанию используются базы данных Microsoft Access (файлы типа .mdb). Далее мы будем рассматривать, если не будет сказано иное, только этот тип баз данных и не будем упоминать данное свойство.

Упомянем ряд свойств, которые управляют режимами работы с Recordset:

- ReadOnly – устанавливает режим "только чтение".
- EOFAction – обработка события "EOF", т.е. события окончания файла. При наступлении события EOFAction=Add New.
- BOFAction – обработка события "BOF".

Свойства DataBaseName и RecordSource элемента управления Data можно устанавливать динамически, используя для ввода значений различные элементы интерфейса, например TextBox. Однако при этом следует для установления связи с новой базой данных и/или таблицей применять метод Refresh, напомним, что при обращении к методам следует указывать объект, и, через точку название метода, например:

Data1.Refresh

Использование стандартных элементов интерфейса для отображения информации таблиц

Элементы интерфейса Picture, Label, TextBox, CheckBox, Image, OLE, ListBox и ComboBox могут быть привязаны к полю данных **Recordset**, управляемому элементом **Data**. Для этого следует установить значения свойств.

DataSource = [имя элемента интерфейса DATA]

DateField = [имя поля].

Дополнительные элементы интерфейса

- Дополнительные стандартные элементы интерфейса полезны как при работе с базами данных, так и во многих других случаях. Получить доступ к ним можно после добавления соответствующих иконок в окно ToolBox.
- Для этого следует открыть диалоговое окно Components, нажав CTRL+T или, воспользовавшись меню Project – Components, в закладке Controls установить флажок напротив требуемого компонента и нажать кнопку “Ok”. Например, чтобы добавить в ToolBox DBGrid, следует отметить “Microsoft Data Bound Grid Control”.

Элемент интерфейса DBGrid

- Grid в переводе с английского означает “решетка”. Элемент **DBGrid** дает адекватное интуитивным представлениям о таблицах баз данных их отображение, позволяет отображать несколько записей **Recordset**, причем достаточно гибким образом, а также требуемое количество столбцов и передвигаться по базе данных как “вверх-вниз” по записям, так и “влево - вправо” по полям.
- Для привязки к данным используется свойство DataSource, в котором следует выбрать из списка одно из имен элементов Data.
- С целью наиболее полного редактирования записей таблицы следует установить свойства AllowAddNew (позволять добавлять новые записи) и AllowDelete (позволять удалять записи) равными True.
- Каждый столбец данных может иметь свое имя, позицию, ширину, вне зависимости от его свойств в физической таблице. Установить заголовок и вид вывода “решетки с данными”, порядок вывода столбцов, их заголовки и прочие характеристики можно при помощи так называемого “Property page” – окна, которое выводится на экран при обращении к свойству Custom.
- Заметим, что решетка существенно привязывается к источнику данных, устанавливать вид вывода данных следует только после задания DataSource, если при работе источник данных изменился (другая таблица или другая структура), следует нажать правую кнопку мыши на элементе **DBGrid** и выполнить команду Retrieve Fields (восстановить поля), выбрав в появляющемся меню соответствующую строку.

Элементы интерфейса DBCombo, DBList

Для использования этих дополнительных элементов следует подключить компонент “Microsoft Data Bound List Control” (см. подключение DBGrid).

Данные элементы позволяют выводить значение установленного поля некоторого набора записей (Recordset) по всем записям. Кроме того, с помощью элементов можно устанавливать непосредственную связь между данными двух наборов записей.

Для вывода списка, сформированного из значений поля в наборе записей, следует установить свойства группы List: RowSource (имя элемента Data) ListField (имя поля).

Для связи двух наборов записей, кроме того, следует установить следующее:

- DataSource – имя управляющего элемента Data.
- RowSource – имя управляемого элемента Data, поле из которого будет использовано для вывода информации.
- DataField – имя поля, значение которого будет передаваться (поле принадлежит элементу набора записей, установленному свойством DataSource).
- BoundColumn – имя поля из набора записей, установленного свойством RowSource. Используется для установления однозначного соответствия между двумя наборами записей. При изменении текущей записи в RowSource, указанному в DataSource производится поиск записи в RowSource, такой, значения полей, установленных в DataField и BoundColumn совпадают.
- ListField – имя поля из набора записей, установленного свойством RowSource, значение которого будет указываться в поле вывода DBCombo.

Приведем пример задания свойств DBCombo. В качестве управляющей таблицы используем **Books**, управляемой – **Authors**, содержащиеся в базе данных Biblio.mdb, поставляемой в качестве примера с системой VB. Обычное место расположения этой базы данных.