Содержание

Введение

1. Основные понятия баз данных

1.1 Базы данных и системы управления базами данных

1.2 Реляционная модель баз данных

2. Проектирование и создание реляционной базы данных для ресторана «Дельфин»

2.1 Построение информационно логической модели данных

2.2 Разработка структур таблиц базы данных и схемы данных

2.3 Описание создания базы данных

3. Создание Web-узла для ресторана «Дельфин»

Заключение

Список использованной литературы

Приложение

Введение

Access — это, прежде всего, система управления базами данных (СУБД). Как и другие продукты этой категории, она предназначена для хранения и поиска данных, представления информации в удобном виде и автоматизации часто повторяющихся операций (таких, как ведение счетов, учет, планирование). С помощью Access можно разрабатывать простые и удобные формы ввода данных, а также осуществлять обработку данных и выдачу сложных отчетов.

Access — мощное приложение Windows; впервые производительность СУБД органично сочетается с теми удобствами, которые имеются в распоряжении пользователей Microsoft Windows. Можно вырезать, копировать и вставлять данные из любого приложения Windows в Access и наоборот; можно создать проект формы в Access и вставить его в конструктор форм.

С помощью объектов OLE (Object Linking and Embedding — связывание и внедрение объектов) можно превратить Access в настоящую операционную среду баз данных. С помощью расширений для Internet можно создавать формы, которые будут напрямую взаимодействовать с данными из World Wide Web, и транслировать их в представление на языке HTML, обеспечивающее работу с такими продуктами, как Internet Explorer и Netscape Navigator.

Базы данных – это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой работают тысячи людей по всей стране. От информации, которая содержится в некоторых базах, может зависеть благополучие множества людей. Поэтому целостность содержимого базы не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файлы перед выключением компьютера, ни от перебоев в электросети.

Цель данной работы – спроектировать и создать базу данных в среде MS Access и Web-узел для Молодежного Центра.

Для достижения данной цели были поставлены следующие задачи:

1) Рассмотреть основные понятия о базах данных;

2) Спроектировать и построить базу данных и Web-узел для Молодежного центра.

Работа состоит из введения, тех глав, заключения списка использованной литературы.

1. Основные понятия баз данных

###

### 1.1 Базы данных и системы управления базами данных

База данных – это организованная структура, предназначенная для хранения информации. В современных базах данных хранятся не только данные, но и информация.

Это утверждение легко пояснить, если, например, рассмотреть базу данных магазина по продаже туров. В ней есть все необходимые сведения о клиентах, об их адресах, видах путевок, их стоимости и т.д. Доступ к этой базе данных имеется у достаточно большого количества сотрудников фирмы, но среди них вряд ли найдется такое лицо, которое имеет доступ ко всей базе полностью и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием базы данных тесно связано понятие системы управления базой данных. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнение ее содержимым, редактирование содержимого и визуализации информации. Под визуализацией информации базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройства вывода или передачи по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем СУБД Microsoft Access, входящую в пакет Microsoft Office.

В состав Базы данных входят:

Язык описания данных (ЯОД) – Средства описания данных в БД и связей между ними. Средствами этого языка описывается структура БД, форматы записей, пароли, защищающие данные.

Язык манипулирования данными (ЯМД) – язык для выполнения операций над данными, позволяющий менять их строение.

Для различных СУБД реализация этих уровней языков может быть различной. В одних случаях ЯОД и ЯМД требует составления пользователем программы полностью “вручную”, в других (что отражает современную тенденцию) в СУБД присутствует средства визуальной (зримой, наглядной) разработки программ. Для этого в современных СУБД имеются редакторы экранных форм, отчетов. “Кирпичиками” (инструментами) таких редакторов являются поля различных видов (поля ввода, поля вывода, вычисляемые поля), процедуры обработки различных типов (формы ввода, таблицы, отчеты, запросы). На основании созданных пользователем объектов программы – генераторы формируют программный код на языке конкретной машины или на промежуточном языке.

Базы данных могут содержать различные объекты. Основными объектами любой базы данных являются ее таблицы. Простейшая база данных имеет хотя бы одну таблицу. Соответственно, структура простейшей базы данных тождественно равна структуре ее таблицы.

Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в простейшей базе данных являются поля и записи. Если записей в таблице пока нет, значит, ее структура образована только набором полей. Изменив состав полей базовой таблицы (или их свойства), мы изменяем структуру базы данных и, соответственно, получаем новую базу данных.

Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access.

Имя поля – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).

Тип поля – определяет тип данных, которые могут содержаться в данном поле.

Размер поля – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.

Формат поля – определяет способ форматирования данных в ячейках, принадлежащих полю.

Маска ввода – определяет форму, в которой вводятся данные а поле (средство автоматизации ввода данных).

Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).

Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

Условие на значение – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных.

Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы.

Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).

Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значение в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видео клипов и других объектов OLE, большинство вышеуказанных свойств не имеют смысла.

Таблицы баз данных, как правило, допускают работу с гораздо большим количеством разных типов данных. Так, например, базы данных Microsoft Access работают со следующими типами данных.

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

Числовой – тип данных для хранения действительных чисел.

Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он храниться в другом месте базы данных, а в поле храниться указатель на него, но для пользователя такое разделение заметно не всегда.

Дата/время – тип данных для хранения календарных дат и текущего времени.

Денежный - тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.

Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

Логический - тип для хранения логических данных (могут принимать только два значения, например Да или Нет).

Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск броузера и воспроизведение объекта в его окне.

Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать их из раскрывающегося списка.

Базы данных – это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями. Выше мы видели, что всю работу по обслуживанию файловой структуры берет на себя операционная система. Для базы данных предъявляются особые требования с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

1.2 Реляционная модель баз данных

В Access в полной мере реализовано управление реляционными базами данных. Система поддерживает первичные и внешние ключи и обеспечивает целостность данных на уровне ядра (что предотвращает несовместимые операции обновления или удаления данных). Кроме того, таблицы в Access снабжены средствами проверки допустимости данных, предотвращающими некорректный ввод вне зависимости от того, как он осуществляется, а каждое поле таблицы имеет свой формат и стандартные описания, что существенно облегчает ввод данных. Access поддерживает все необходимые типы полей, в том числе текстовый, числовой, счетчик, денежный, дата/время, MEMO, логический, гиперссылка и поля объектов OLE. Если в процессе специальной обработки в полях не оказывается никаких значений, система обеспечивает полную поддержку пустых значений.

Реляционная обработка данных в Access за счет гибкой архитектуры системы способна удовлетворить любые потребности. При этом Access может использоваться как автономная СУБД в режиме файл-сервера или клиентского компонента таких продуктов, как SQL Server. Кроме того. Access поддерживает протокол ODBC (Open Database Connectivity), что позволяет подключаться к базам данных множества различных форматов, таких как SQL Server, Oracle, Sybase и даже DB/2 для больших ЭВМ фирмы IBM.

Система Access поддерживает обработку транзакций с гарантией их целостности. Кроме того, предусмотрена защита на уровне пользователя, что позволяет контролировать доступ к данным отдельных пользователей и целых групп.

Мастер (Wizard) может превратить часы работы в считанные минуты. Мастера задают наводящие вопросы относительно содержания, стиля и формата создаваемого объекта; затем они автоматически строят нужный объект. В составе Access около ста мастеров, помогающих конструировать базы данных, приложения, таблицы, формы, отчеты, диаграммы, почтовые наклейки, элементы управления и свойства. Допускается даже настройка мастеров для решения разных задач.

Окна конструкторов форм и отчетов имеют одинаковый интерфейс и предоставляют пользователю много возможностей. Форма или отчет конструируется по принципу WYSIWYG (What You See Is What You Get — что видишь, то и получишь). Добавляя очередной элемент управления, пользователь видит, как при этом изменяется создаваемая форма.

В формы и отчеты можно включать надписи, поля текстовых данных, переключатели, флажки, линии и прямоугольники, а также оформлять их, выделяя элементы цветом и тенью. Более того, можно включать целые рисунки, диаграммы, подформы и подотчеты. При этом все параметры представления данных остаются полностью подконтрольными пользователю. Формы могут занимать много страниц, а в отчетах может быть предусмотрено много уровней группировки данных и подведения итогов.

Формы и отчеты можно просматривать в режиме предварительного просмотра, обеспечивая взгляд "с высоты птичьего полета" путем изменения масштаба. В режиме конструирования отчет можно просматривать с фиктивными данными, чтобы не дожидаться обработки большого реального файла.

Конструктор отчетов — очень мощное средство, допускающее использование до десяти уровней группировки и сортировки. Благодаря ему существует возможность создания отчетов, демонстрирующих процентные и итоговые показатели, получить которые можно лишь за два прохода. Допускается создание многих типов отчетов, которые включают почтовые наклейки и списки рассылки почты.

Можно создавать запросы, которые обеспечивают вычисление итогов, отображение сгруппированных и построение новых таблиц. Запрос можно использовать даже для обновления данных в таблицах, удаления записей и добавления одной таблицы к другой.

2. Проектирование и создание реляционной базы для Молодежного Центра

2.1 Построение информационно логической модели данных

Работа по созданию базы данных началась с карандашом и листом бумаги в руках. На данном этапе компьютер просто не нужен. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, впоследствии очень трудно устраняются, поэтому этот этап является основополагающим.

База данных будет создана для кафе Дельфин. Основной его деятельностью является общественное питание. Таким образом, в базе данных нужно учесть поступление и расход продуктов.

На компьютере работу началась с составления генерального списка полей. В соответствии с типом данных, размещаемых в каждом поле, был определен наиболее подходящий тип для каждого поля.

В каждой из таблиц наметили ключевое поле. В качестве такого было выбрано поле, данные в котором повторяться не могут. Если в таблице вообще нет ни каких полей, которые можно было бы использовать, как ключевые, всегда можно ввести дополнительное поле типа Счетчик – оно не может содержать повторяющихся данных по определению.

По ходу разработки проекта непременно будут приходить в голову новые идеи. Если схема данных составлена правильно, подключать к базе новые таблицы нетрудно.

На этом этапе было завершено предварительное проектирование базы данных, и на следующем этапе начинается ее непосредственная разработка. С этого момента следует начать работу с СУБД.

2.2 Разработка структур таблиц базы данных и схемы данных

Выяснив основную часть данных, которые нужно оформить в базе данных, приступили к созданию структуры базы, то есть структуры ее основных таблиц.

В базе создано две таблицы. В первой, которая называется «Поставщики» учтена следующая информация: «Код поставщика», «Наименование поставщика», «Наименование товара», «Дата поставки» и «Код товара».

Рис. 2.2.1. Создание таблицы «Поставщики» в режиме Конструктора

Распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. В каждой из таблиц наметили ключевое поле. В качестве такого выбирают поле, данные в котором повторяться не могут.

Рис. 2.2.2. Создание таблицы «Расход» в режиме Конструктора.

В таблице «Поставщики» назначим ключевым «Код поставщика», в таблице «Расход» - «Код продукта».

Обе таблицы связаны через поле «Код продукта».

Рис. 2.2.3

2.3 Описание создания базы данных

Далее на основе таблиц были созданы запросы

Рис. 2.3.1. Создание запроса на выборку в режиме Конструктора

При создании были указаны данные, которые необходимы в запросе.

Рис. 2.3.2. Вид запроса

Далее приступаем с созданию форм. Формы – одно из основных средств для работы с базами данных в Access - используются для ввода новых записей (строк таблиц), просмотра и редактирования уже имеющихся данных, задания параметров запросов и вывода ответов на них и др. Формы представляют собой прямоугольные окна с размещенными в них элементами управления. Существует возможность создания форм динамически при исполнении программы, однако естественным режимом их создания является режим визуального конструирования. Выбором команды Форма в меню Вставка вывели на экран окно Новая Форма, позволяющее задать таблицу или запрос, для которых создается новая форма, и указали режим ее создания. Кроме создания формы «вручную», создание формы можно автоматизировать, используя Мастер форм (FormWizard).

Рис. 2.3.1. Выбор полей для формы

В качестве исходной таблицы мы выбрали таблицу «Поставщики», в качестве метода создания форм - Мастер форм. После нажатия кнопки Ok, переходим к следующему диалоговому окну

Укажем поля, которые необходимо поместить на форму. Кнопки расположенные в нижней части экрана позволяют возвратиться назад на один шаг или продолжить процесс формирования формы далее.

На следующих шагах Мастер форм предлагает выбрать форму представления отчета (в столбец, ленточную, табличную или выровненную) и стиль оформления. Стили представляют собой набор различных фоновых рисунков с соответствующим подбором шрифтов и форм полей.

Рис. 2.3.2. Выбор макета формы

На последнем шаге изменим предлагаемое Мастером название формы на свое собственное и на этом завершим процесс создания формы нажав кнопку Готово.

Рис. 2.3.3. Вид формы «Поставщики»

Форма для таблицы «Расход» создается аналогично.

Рис. 2.3.4. Форма «Расход»

Отчеты, как и формы, являются главными элементами интерфейса баз данных. Они позволяют выводить на печать информацию о содержимом базы в удобном для пользователя виде.

Мы создаем отчет с помощью Мастера для распечатки списка клиентов. В качестве исходных данных выберем запрос Поставщики.

Рис. 2.3.3. Выбор полей для Отчета

На следующем шаге Мастер отчетов предлагает осуществить группировку данных. После чего Мастер отчетов предлагает ввести сортировку.

На следующих шагах был выбран макет отображения отчета и ориентацию листа бумаги, а также определены стиль заголовка и подножий.

Рис. 2.3.5. Выбор макета отчета

Шагнув далее, мы увидели на экране окно диалога с клетчатым флагом, который указывает на то, вы подошли к финишу. Зададим наименование отчета и нажмем кнопку Готово.

Рис. 2.3.6. Выбор названия отчета

Модифицируем отчет следующим образом: выберем его имя на вкладке Отчеты и щелкнем на кнопке Конструктор. Выбор команды Просмотр в том же окне позволяет увидеть, как будет выглядеть распечатанный отчет.

Рис. 2.3.7. Вид отчета.

3. Создание Web-узла для Молодежного Центра

Создание узла Интернета или интрасети должно быть разделено на две основные задачи. Первая - создание наполнения. Сюда входит разработка и оформление материалов и ресурсов, которые человек хочет сделать доступными для какой-либо аудитории. Для преобразования подобной информации используется простой язык программирования. Вторая задача - публикация этих ресурсов в сети.

Мы заполнили сервер Web состоящий из трех основных частей: проектирования узла, программирования и размещения его содержимого на сервере. В начале необходимо рассмотреть формат и структуру узла и определить наилучший способ представления информации. Составив общий план, мы решили, как привести информацию к форме, доступной программам просмотра Web. Язык программирования, используемый для этого, называется HTML (Hypertext Markup Langu-age). И, наконец, был выбран оптимальный способ размещения информации на сервере.

По существу, узел Web состоит из одной или более страниц Web и одной или более домашних страниц. Страницы Web - это организованное собрание материалов узла Web; они могут быть связаны друг с другом. Домашняя страница - это вводная или главная страница узла.

Во многих отношениях страница Web напоминает печатную страницу. Это - набор слов, фраз, графики и других элементов, организованных в виде последовательного потока данных от начала страницы до ее конца. Самое большое различие страниц в том, что длина страницы Web не имеет физических ограничений. Кроме того, страницы Web содержат связи, соединяющие их между собой. Это означает, что их не надо просматривать в некотором определенном порядке или последовательности, как страницы большинства книг.

Домашняя страница служит точкой входа в узел Web. Обычно это вводная страница, содержащая приветствие и ссылки на страницы второго уровня, доступные на данном узле. Домашние страницы формируют первое впечатление, которое производит узел на пользователя. Некоторые маленькие узлы Web типа персональных страниц состоят только из домашней страницы, тогда как на больших узлах может быть несколько домашних страниц.

Узел Web. Организация материалов на узле WWW в чем-то похожа на написание книги или статьи. Вначале необходимо продумать структуру и формат, а также ответить на множество вопросов. О чем следует сказать в первую очередь? Как говорить об этом? Какие темы логически связаны или дополняют друг друга? Каков порядок изложения? Нужны ли рисунки или иная графика? В зависимости от объема и вида информации реализация этих вопросов может оказаться простой, а может и весьма трудной.

Организация информации. Диапазон способов организации узлов Web весьма широк: от узлов, имеющих строгую линейную структуру, до узлов, у которых вообще нет четкой структуры. Обычно страницы располагают в иерархическом или линейном порядке, а также в виде паутины. Вместо того, чтобы с самого начала жестко структурировать информацию надо убедиться, что связи между страницами - логичные, что они ведут пользователя к необходимой ему информации. Вероятно, для этого лучше всего подойдет некоторая комбинация различных способов организации информации.

Иерархическая организация. Узлы Web, которые следуют иерархической или древовидной организации, имеют единственную точку входа в узел; остальные страницы располагаются на исходящих из нее ответвлениях. Данный подход удобен, если можно легко разбить свою информацию на категории и подкатегории. При иерархической организации узла Web к странице самого нижнего уровня ведет один и только один путь.

Такая строгая структура узла может вызвать проблемы у пользователей. Например, если пользователь прошел на несколько уровней вниз по одному из путей, а потом решил попасть в другую часть дерева, то ему придется возвращаться обратно.

Линейная организация. Чтобы пользователи пользователи узла Web могли читать содержимое узла как книгу или журнал или чтобы они прошли по за- данному пути от начала и до конца узла - необходимо выбрать линейную организацию. Какая-то страница может иметь несколько связей с примечаниями и дополнениями, но для продвижения дальше пользователь должен вернуться на нее снова. Продвижение по документу осуществляется кнопкой Next, а возврат к началу узла - кнопкой Prev.

Для большого узла Web линейная организация подходит не очень. Читателям, ищущим конкретную информацию, может не понравиться необходимость пройти через множество страниц, прежде чем они попадут на нужную. Используя линейный подход, не надо поддаваться искушению создавать слишком длинные страницы Web: многие люди найдут просмотр их очень утомительным.

Организация информации в виде паутины, вероятно, наилучшим образом подходит для большинства случаев. В этой структуре страницы связаны друг с другом общим контекстом. К одной странице может вести несколько связей, и у каждого документа есть, по крайней мере, два выхода. Связи иногда образуют круг. Для путешествий по Web эта организация узла наиболее привлекательна. Быстрый проход по узлу и большая свобода перемещения - вот что нравится пользователям. Главный недостаток такой структуры в том, что пользователь может что-нибудь пропустить, так как он может пройти не по всем связям.

Заключение

В Microsoft Access для обработки данных базовых таблиц используется мощный язык SQL (структурированный язык запросов). Используя SQL можно выделить из одной или нескольких таблиц необходимую для решения конкретной задачи информацию. Access значительно упрощает задачу обработки данных. Совсем не обязательно знать язык SQL. При любой обработке данных из нескольких таблиц Access использует однажды заданные связи между таблицами.

В Microsoft Access имеется также простое и в то же время богатое возможностями средство графического задания запроса – так называемый «запрос по образцу» (query by example), которое используется для задания данных, необходимых для решения некоторой задачи. Используя для выделения и перемещения элементов на экране стандартные приемы работы с мышью в Windows и несколько клавиш на клавиатуре, можно буквально за секунды построить довольно сложный запрос.

Microsoft Access спроектирован таким образом, что он может быть использован как в качестве самостоятельной СУБД на отдельной рабочей станции, так и в сети – в режиме «клиент-сервер». Поскольку в Microsoft Access к данным могут иметь доступ одновременно несколько пользователей, в нем предусмотрены надежные средства защиты и обеспечения целостности данных. Можно заранее указать, какие пользователи или группы пользователей могут иметь доступ к объектам (таблицам, формам, запросам) базы данных. Microsoft Access автоматически обеспечивает защиту данных от одновременной их корректировки разными пользователями. Access также опознает и учитывает защитные средства других подсоединенных к базе данных структур (таких, как базы данных Paradox, dBASE и SQL).

Практически все существующие СУБД имеют средства разработки приложений, которые могут использованы программистами или квалифицированными пользователями при создании процедур для автоматизации управления и обработки данных.

Microsoft Access предоставляет дополнительные средства разработки приложений, которые могут работать не только с собственными форматами данных, но и с форматами других наиболее распространенных СУБД. Возможно, наиболее сильной стороной Access является его способность обрабатывать данные электронных таблиц, текстовых файлов, файлов dBASE, Paradox, Btrieve, FoxPro и любой другой базы данных SQL, поддерживающей стандарт ODBE. Это означает, что можно использовать Access для создания такого приложения Windows, которое может обрабатывать данные, поступающие с сетевого сервера SQL или базы данных SQL на главной ЭВМ.

Список использованной литературы

1. Вейскас Д., Эффективная работа с Microsoft Access 2, С.-Пб.,2005
2. Биллиг В.А., Дехтярь М.И., VBA и Office 97 Офисное программирование, М., изд. «Русская редакция», 2004
3. Гусева Т.И., Башин Ю.Б. , Проектирование баз данных в примерах и задачах, М.,2002
4. Савельев В.А., Персональный компьютер для всех (кн.3), Создание и использование баз данных, М.,2003
5. Хоффбауэр М., Шпильманн К., ACCESS, Сотни полезных рецептов, Киев, «BHV», 2002
6. Тимоти Бадд, Объектно- ориентированное программирование в действии, С.-Пб., 2004
7. Росс Нельсон ,Running Visual Basic for Windows, М., изд. «Русская редакция», 2005
8. Бемер С, Фратер Г., MS Access … для пользователя, Киев, «BHV», 2004,
9. Винтер Рик, Microsoft Access 97, Справочник, С.-Пб., «Питер», 2002

Приложение 1

Пошаговое создание Web-страницы

Рис. 1. Открытие мастера

Рис. 2. Задание требуемого порядка сортировки

Рис.3. Выбор названия страницы