**Псковский Вольный университет**

**Курсовая работа**

**MICROSOFT ACCESS**

**студетов 3 курса**

**факультет информационных систем**

**Банахова Олега Святославовича**

**Зуева Юрия Владимировича**

**Научный консультант: Басин Юрий Абрамович.**

**г.Псков 1995г.**

**СОДЕРЖАНИЕ**

1. **Введение.**
2. **Access - что это такое.**

**1. Несколько слов о ...**

**2. Таблицы.**

**3. Запросы.**

**4. Формуляры.**

**5. Отчеты.**

**6. Макросы.**

**7. Модули.**

1. **Построение небольшой Базы данных.**
2. **Небольшая сравнительная характеристика.**

**1. ВВЕДЕНИЕ.**

Перед тем как приступить к изучению **СУБД ACCESS** рассмотрим несколько основных понятий.

***Данные*** - систематизированная и структурированная каким-либо образом информация. Существуют по крайней мере три модели данных: ***иерархическая***, ***сетевая***, ***реляционная***.

***Модели данных*** - представление данных и их взаимосвязей, описывающих понятия проблемной среды. Модели данных используются как для концептуального , так и для логического и физического представления данных.

В данной работе мы будем рассматривать только логическое представление данных, т.е. с точки зрения прикладного программиста и пользователя. С их точки зрения данные делятся на записи, те в свою очередь делятся на поля, и между полями, а так же и между записями могут быть установлены различные связи.

***Поле*** - *основной и минимальный элемент данных*.

***Запись*** - *несколько полей*.

***Связи*** - *логические взаимосвязи между записями или полями*.

Иерархически структурированные данные состоят из нескольких записей одна из которых определена как корневая. Примером иерархической модели могут служить деревья:

Где *КЛИЕНТ*, *ЗАКАЗ*, *ТОРГОВЫЙ АГЕНТ*  являются записями (*КЛИЕНТ* - корневая запись), *НОМЕР*, *ФАМИЛИЯ*, *ИМЯ*, *ОТЧЕСТВО* - поля записи *КЛИЕНТ*, линиями показаны связи.

***Дерево*** - *ориентированный граф(граф - пара множеств, одно из которых описывает множество вершин, а другое множество связей между ними) где все вершины, кроме корня, находятся в голове только одной дуги, корень не находится в голове ни одной из дуг и связан с вершиной дерева*.

Между записями в иерархии могут быть определены связи: “*один ко* *многим*”, или “*один к одному*”, где запись соответствующая элементу “*один*” указанной связи, определяется как исходная, а соответствующая элементу “*много*” - как порожденная.

Сетевая модель подобна иерархической, но является более общей моделью, в том смысле, что между записями присутствуют связи “*много* *ко многим*” и “*много к одному*”, а так же здесь нет корневого узла, т.е. любая запись может быть корневой.

***Сетевая модель***:

Где *КЛИЕНТ*, *ЗАКАЗ*, *ТОРГОВЫЙ АГЕНТ*  являются записями, *НОМЕР*, *ФАМИЛИЯ*, *ИМЯ*, *ОТЧЕСТВО* - поля записи *КЛИЕНТ*, линиями показаны связи.

Реляционная модель данных является одним из самых простых видов представления информации и самой легкой концепцией для понимания. Эта модель представляет собой данные упорядоченные в таблицы, чаше всего двухмерные. Таблицы такого вида называются ***отношениями*** (*реляциями*) и каждая таблица представляет собой отдельный файл.

Реляционная модель:

**Строка Столбец**

|  |  |  |  |
| --- | --- | --- | --- |
| **НОМЕР - КЛИЕНТА** | **ФАМИЛИЯ - КЛИЕНТА** | **ИМЯ - КЛИЕНТА** | **ОТЧЕСТВО - КЛИЕНТА** |
|  |  |  |  |
|  |  |  |  |

**Поле**

|  |  |  |
| --- | --- | --- |
| **НОМЕР - ЗАКАЗА** | **НАЗВАНИЕ - ЗАКАЗА** | **ТОРГОВЫЙ АГЕНТ** |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НОМЕР - ТОРГОВОГО АГЕНТА** | **ФАМИЛИЯ** | **ИМЯ** | **ОТЧЕСТВО** | **ПОСТАВЛЯЕМЫЙ ТОВАР** |
|  |  |  |  |  |
|  |  |  |  |  |

Каждая таблица состоит из столбов, строк каждому столбцу присвоено однозначное имя, один элемент столбца называется ***полем***, а строка называется ***записью***. Данный пример показывает, что и сетевая модель, и иерархическая могут быть приведены к реляционной.

Структурированные таким образом данные могут хранится в ЭВМ в виде ***Баз Данных***. База данных может быть определена как совокупность предназначенных для машинной обработки и хранения данных, которые могут использоваться одним или несколькими пользователями.

В зависимости от рассмотренных выше моделей данных базы данных могут быть иерархическим, сетевыми и реляциоными. Дальше мы будем рассматривать только реляционные базы данных.

С базами данных неразрывно связано такое понятие как ***СУБД***. ***СУБД***(*система управления базами данных*) - программа которая управляет данными, осуществляет хранение, извлечение, поиск, редактирование информации хранимой в базе данных. СУБД также подразделяются на иерархические, сетевые и реляционые в зависимости от данных которые они обрабатывают. Существуют множество различных СУБД: **IMS**, **СЕТОР**, **ORACLE**, **CLIPPER**, **FOX PRO**, **ACCESS**. В данной работе мы рассмотрим только ***MICROSOFT*** ***ACCESS***.

**2. MICROSOFT ACCESS.**

**Несколько слов о ...**

***Microsoft Access*** - это интерактивная реляционная СУБД(*relational database management system - RDBMS* ) для ***WINDOWS*** . Это программа, которую Вы можете использовать для хранения и извлечения данных в зависимости от отношений, которые Вы установили. Работа с ней упрощена посредством манипулятора мыши. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток. Все это благодаря поддержки ***True-type*** шрифтов и встраивания ***OLE-объектов***(*Object Linking and Embeding*) в рамках среды ***WINDOWS***. ***OLE - объект*** представляет собой ссылку на определенную информацию, которая остается в своей первоначальной форме. ***OLE-объектом*** может быть ***EXCEL - таблица***, ***Paintbrush - иллюстрация*** или ***Sound - файл.***

Новинкой программы ***Access*** является ***Cue Cards*** - Система суфлирования. Это обучающая система, предоставляющая пользователю рекомендации по выходу из реальных ситуаций при решении прикладных задач. При знании английского языка пользователь всегда сможет получить подсказку о том, что следует делать далее.

***Access*** также предоставляет в распоряжение пользователя механизмы работы с базами данных различных форматов. К примеру, можно прямо обращаться к базам данных ***dBASE***, ***Paradox*** или ***Btrieve*** без конвертирования их в формат, используемый ***Access***. В состав пакета ***Access*** также входит язык ***Access Basic*** (*встроенный диалект языка Visual Basic* ), дающий возможность формирования специализированных систем управления базами данных.

Теперь перейдем к более подробному рассмотрению системы ***Access***. Основными понятиями или объектами этой системы являются: ***таблицы***, ***запросы***, ***формуляры***, ***отчеты***, ***макросы*** и ***модули***. И конечно же главным понятием будет ***база данных*** (*database*).

***База данных*** это набор данных, который связан с определенной темой или назначением. Когда Вы создаете базу данных в Microsoft Access, Вы создаете один файл, который содержит все таблицы, запросы, формуляры, отчеты и другие объекты, которые помогают Вам использовать информацию в базе данных. Для создания базы данных необходимо выполнить следующие шаги :

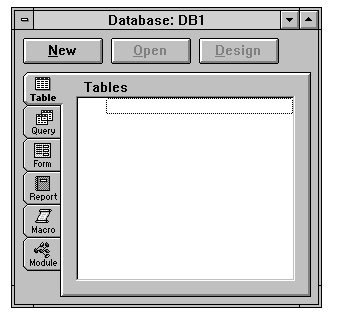
* Активизируйте окно ***Microsoft Access*** и выберите в меню ***File*** директиву ***New Database*** (*или щелкните кнопкой New Database в* *строке пиктограмм****). Microsoft Access*** высвечивает диалоговое окно ***New Database*** и автоматически дает имя Вашей базе данных , которое Вы можете заменить.
* В строке ***File Name***, напечатайте Ваш вариант имени базы данных. Оно может содержать до 8 символов без пробелов. ***Microsoft Access*** автоматически добавляет расширение *.MDB* к имени Вашей базы данных, если Вы этого не сделали.

Если Вы хотите хранить базу данных в определенном месте выберите соответствующую директорию в списке директорий ***Directories list***.

Нажмите на кнопку ***OK***.

***Microsoft Access*** создает пустой файл базы данных и открывает окно базы данных ***Database window***.

Окно Базы данных:



Созданную базу данных можно наполнить объектами различного рода и выполнять операции с ними. Но с базой данных можно выполнять операции как с неделимым образованием. Все операции такого рода - операции управления базой данных - сосредоточены в меню ***File*** прикладного окна ***Access*** или в окне базы данных. Например, операция открытия базы данных включает следующие шаги:

Если необходимо, активизируйте ***Microsoft Access***.

Из меню ***File***  выберите директиву ***Open Database*** (*или щелкните кнопкой Open Database в строке пиктограмм* ). ***Microsoft*** ***Access***  высвечивает диалоговое окно ***Open Database***.

В строке ***File Name*** выберите соответствующее имя базы данных, предварительно выбрав соответствующую директорию в списке директорий ***Directories list***  и соответствующий накопитель в строке ***Drives***.

Нажмите на кнопку ***OK***.

***Microsoft Access*** высветит окно соответствующей базы данных. При открытии базы данных можно ограничить возможные операции с базой данных только чтением и ограничить круг пользователей, имеющих доступ к базе данных только собственной персоной. Такие ограничения могут оказаться полезными, если база данных эксплуатируется в многопользовательской системе. Если необходимо только просмотреть базу данных и содержащиеся в ней объекты, но не изменять их, то при открытии базы данных необходимо активизировать контрольный индикатор ***Read Only*** (*только чтение*).

Если необходимо предотвратить изменения данных и объектов данной базы данных со стороны других пользователей, то при открытии базы данных необходимо установить опцию ***Exlusive*** (*исключительные права*).

Базу данных можно подвергнуть сжатию, для этого необходимо выбрать в меню ***File*** директиву ***Compact Database*** (*сжать базу* *данных*). ***Access*** создает копию базы данных и по-новому размещает ее на диске. После выбора сжимаемого файла в диалоговом окне сжатия базы данных появляется диалоговое окно цели сжатия(***Database to Compact Into***), в котором можно выбрать имя для сжатой версии базы данных. Если используется то же самое имя, и база данных оказывается успешно сжатой, то по завершении сжатия ***Access*** автоматически уничтожает исходную(несжатую) версию.

Базу данных можно зашифровать для защиты секретной информации от просмотра. Это осуществляется с помощью директивы ***Encrypt Database*** (*расшифровать* - *Decrypt Database*). В появившемся диалоговом окне выберите подлежащую шифрованию/дешифрованию базу данных и щелкните на кнопке ***OK***. В результате откроется диалоговое окно целевого файла, где следует указать имя файла, в который будет помещен результат шифрования/дешифрования. Если используется то же самое имя и база данных была успешно зашифрована/расшифрована, то  ***Access*** автоматически заменяет оригинальный файл на полученный в результате преобразования.

В некоторых ситуациях база данных может оказаться поврежденной. Для восстановления базы данных служит директива ***Repair Database*** (*восстановить базу данных*) в меню ***File***. Эту операцию рекомендуется применять в целях профилактики так как ***Access*** не замечает, что база данных была повреждена. В диалоговом окне ремонта нужно ввести имя, директорию и накопитель поврежденной(*или подозреваемой в повреждении*) базы данных и нажать кнопку OK. После этого ***Access*** займется “починкой” заданной базы данных и спасет то, что еще можно спасти.

Если пользователь закончил работу с базой данных, то ее следует закрыть. Для этого:

Активизируйте окно базы данных, если это необходимо.

В меню ***File*** выберите директиву ***Close Database***.

Можно закрыть базу данных и с помощью двойного щелчка мышью на кнопке вызова управляющего меню окна базы данных или нажатием клавиш (*Ctrl + F4*).

**Таблицы.**

Следующим шагом в построении базы данных будет ее заполнение, т.е. создание объектов и присвоение последним необходимых свойств. Какого бы типа ни были Ваши данные, Вам придется хранить их в одной или нескольких таблицах. Таким образом, первым шагом в построении базы данных будет создание таблицы.

***ТАБЛИЦА*** - это набор записей(строк таблицы), состоящих из отдельных полей(колонок таблицы), в которых хранится информация и составляющая, собственно, содержание базы данных.

Таблицы являются основной формой представления информации, содержащейся в базе данных. Без таблицы нельзя спроектировать формуляр, на базе таблиц составляются запросы и отчеты. Создать таблицу можно двумя способами:

С помощью ***Конструктора таблиц***.

В окне базы данных щелкните на кнопке ***Table***.

Выберите кнопку ***New*** в окне базы данных.

***Microsoft Access*** высветит диалоговое окно ***New Table***.

Выберите кнопку ***TableWizards***(*Конструктора таблиц*).

В диалоговом окне ***Конструктора*** следуйте указаниям по добавлению полей в Вашей таблице.

Без помощи ***Конструктора***.

В окне базы данных щелкните на кнопке ***Table*** (*или выберите* *директиву* ***Tables*** *в меню* ***View***).

Нажмите кнопку ***New*** в окне базы данных.

***Microsoft Access*** высветит диалоговое окно ***New Table***.

Нажмите кнопку ***New Table***.

***Microsoft Access*** откроет пустое окно проектирования таблицы. Оно будет выглядеть так:

В этом окне задаются характеристики полей, составляющих запись (*блок данных*) базы, т.е. вводится спецификация записи. В нижней части окна ***Access*** выдает справочную информацию о допустимых на каждом шаге проектирования действиях.

Каждому полю, включаемому в запись, приписывается тип данных, определяющий вид информации, которая будет храниться в данном поле. Тип данных вносится в колонку ***Data Type***, причем его можно выбрать из списка доступных типов.

***Access*** различает следующие типы данных полей:

***Text*** Текстовые поля содержат текст, длина которого не может быть более 255 символов. Реальная длина поля устанавливается с помощью параметра ***Field Size***(*размер* *поля*).

***Memo*** В Memo-полях сохраняется текст длиной до 32000 символов. Поля этого типа данных не могут индексироваться.

***Number*** Числовые поля содержат произвольные числовые значению. Диапазон допустимых значений определяется параметром ***Field Size*** (*размер поля*).

***Date/Time*** Поля даты/времени содержат значения даты и времени в диапазоне от 100 до 9999 года.

***Currency*** В денежных полях можно хранить числа с точностью до 15 разрядов слева от запятой и четырех десятичных разрядов справа от запятой.

***Counter*** Поле счетчика содержит число, которое автоматически увеличивается ***Access*** на 1, когда в таблицу добавляется новый блок данных.

***Yes/No*** В таких полях сохраняются значения Да или Нет. Поля данного типа не могут индексироваться.

***OLE Object*** В *OLE - полях* помещаются объекты, такие как, например, *Excel-таблица* или *Microsoft Draw* - графика, обработанные *OLE - сервером*. Размер поля может быть до 128 МБ. Поля данного типа не могут индексироваться.

Характеристики каждого поля определяются рядом параметров(свойств). Эти параметры объявляют способы обработки, сохранения и индикации данных. Параметры специфицируемого поля перечисляются в режиме проектирования в нижней части окна таблицы. Набор характеризующих поле параметров зависит от типа данных, выбранного пользователем для данного поля.

***Field Size*** Размер поля устанавливает максимальную длину текстового поля или способ представления чисел в поле типа ***Number***.

***Format*** Формат определяет способ представления данных. Наряду с предопределенными форматами допускается использование собственных форматов пользователя.

***Decimal Places*** Десятичные разряды устанавливают количество разрядов справа от десятичной запятой.

***Caption*** Заголовок определяет надпись, которая будет использоваться в качестве наименования поля в формуляре или отчете. Если для этого параметра не будет задано никакое значение, то в качестве надписи будет по умолчанию использовано имя поля.

***Default Value*** Значение по умолчанию устанавливает значение, которое будет автоматически введено в поле при генерации блока данных.

***Validation Rule*** Ограничение ввода - правило, ограничивающее допустимые для ввода в поле данные.

***Validation Text*** Сообщение о нарушении - текст возникающий при попытке ввести в поле данные, не удовлетворяющие правилу, сформулированному в ***Validation Rule***.

***Indexed*** Признак индексирования.

В готовую спецификацию можно вносить изменения. В частности, можно изменять параметры отдельных полей, добавлять поля в запись в нужных местах и удалять лишние. Но при этом следует постараться внести все исправления в спецификацию до начала заполнения базы данных, т.к. попытка изменить параметры полей, заполненной базы может повлечь за собой потерю или искажение данных.

Переупорядочение полей:

Выберите поле, подлежащее переразмещению, щелкнув кнопкой мыши в селекторной колонке, расположенной на левом краю окна таблицы.

Щелкните в селекторной колонке опять и удерживая правую кнопку мыши нажатой перетащите указатель мыши в нужную позицию.

***Microsoft Access*** расположит маркированное Вами поле в новом месте.

Удаление полей.

Откройте таблицу в режиме проектирования.

Маркируйте подлежащее удалению поле, щелкнув кнопкой мыши в селекторной колонке.(*Или используйте клавиши управления курсором и подведя указатель к соответствующему полю нажмите комбинацию клавиш Shift+Spacebar*).

Из меню ***Edit*** выберите директиву ***Delete Row*** (*или нажмите кнопку Delete Row в строке пиктограмм)*.

***Microsoft Access*** удалит поле и все его данные из Вашей таблицы.

Вставка полей.

Установите указатель строки к полю, над которым Вы бы хотели разместить новое поле, с помощью мыши или клавишами управления курсором.

Из меню ***Edit*** выберите директиву ***Insert Row***(*или выберите кнопку Insert Row в строке пиктограмм)*.

***Microsoft Access*** вставит пустое поле над текущей строкой.

***Microsoft Access*** работает наиболее эффективно, если Вы определили первичный ключ. Первичный ключ таблицы состоит из одного или нескольких полей, которые однозначно идентифицируют каждую запись в Вашей таблице. Вам следует всегда определять первичный ключ т.к. при этом таблица имеет следующие преимущества:

***мicrosoft Access*** автоматически создает индекс для первичного ключа. Этот индекс ускоряет процессы запросов и другие операции.

Когда Вы просматриваете записи, ***Microsoft Access*** покажет их упорядоченными по первичному ключу по умолчанию.

Когда Вы добавляете данные к Вашей таблице, ***Access*** не позволит хранить данные с одним и тем же значением первичного ключа. Поэтому Вы можете быть уверены ,что каждая запись уникальна.

Установка первичного ключа.

Маркируйте поле, которое Вы хотите использовать в качестве первичного ключа. Для объявления нескольких полей первичным ключом, удерживая клавишу Ctrl маркируйте все необходимые поля.

В меню ***Edit*** выберите директиву ***Set Primary Key*** (*или нажмите* *кнопку Set Primary Key в строке пиктограмм*).

У полей определенных как первичный ключ слева от строки добавится идентификатор в виде ключа.

В ***Access-таблицы*** можно встроить объекты из других поддерживающих интерфейс ***OLE*** прикладных программ и связать эти объекты с их родительскими программами.

При встраивании объекта ***Access*** сохраняет объект в таблице. Двойным щелчком мышью на объекте запускается прикладная программа, с помощью которой он был изготовлен - в ней можно произвести изменения в объекте. После завершения работы с родительской программой объект в измененной форме будет сохранен в таблице. Объект следует встраивать только в том случае, если его необходимо обрабатывать не покидая ***Access***. Обычно речь идет об объекте , который должен стать фиксированной составной частью базы данных, но его нельзя изготовить прямо в ***Access***, например, ***Paintbrush-иллюстрация*** на странице ***Access-отчета***.

Если же объект, порожденный другой прикладной программой, связывается с ***Access-таблицей*** без встраивания, то объект сохраняется не в таблице, а в отдельном файле, причем формат этого файла определяется прикладной программой, с помощью которой объект был создан. В момент загрузки и открытия таблицы ***Access*** связывается с этим файлом, извлекает его содержимое и вставляет его в таблицу. Связывание объектов (а не встраивание) рекомендуется в тех случаях, когда в ***Access-базу*** необходимо включить объект, не подлежащий изменению при работе с ***Access***. Например, речь может идти о таблице счетов, обрабатываемой ***Excel*** и используемой в нескольких ***Access-базах***. Наличие связи этих баз с одним и тем же ***Excel-файлом*** гарантирует наличие последней версии таблицы счетов во всех этих базах. Преимущество связи заключается в том, что один оригинальный объект можно связать с несколькими ***Access-таблицами***, и при этом его не нужно многократно подвергать сохранению.

Встраивание объектов.

В меню ***Edit*** обратитесь к директиве ***Insert Object*** (*Эта директива будет* *доступна в том случае, если в спецификации таблицы активному в* *данный момент полю был присвоен тип OLE Object*.)

В списке ***Object Type*** выберите необходимую Вам, поддерживающую ***OLE интерфейс*** прикладную программу.

Если объект уже существует, щелкните мышью на кнопке ***File*** , чтобы открыть диалоговое ***окно Insert Object from File***.

Выберите нужный файл и нажмите кнопку ***OK***.

Связывание объектов.

В меню ***Edit*** выберите директиву ***Past Link***. (*Однако эта директива* *становится доступной лишь после того , как подлежащий связыванию* *объект будет скопирован, т.е. помещен в буфер обмена Clipboard* .)

Нажмите кнопку ***OK***.

***Access*** позволяет создавать реляционные базы данных. В реляционной базе данных пользователь может описать отношения между несколькими таблицами. ***Access*** учитывает эти отношения при поиске взаимосвязанных данных во время обработки запросов, формуляров и отчетов, базирующихся на нескольких таблицах.

Установка связи между таблицами.

В меню ***Edit*** выберите директиву ***Relationships***, или щелкните мышью на кнопке ***Relationships*** в строке пиктограмм. Откроется окно ***Relationships*** , затем окно ***Add*** ***Table***.

Если окно ***Add Table*** не появилось, выберите соответствующую директиву в окне ***Relationships.***

В строке ***Table/Query*** выберите таблицу для которой вы хотите создать связь. Для выбора более чем одной таблицы одновременно, удерживая клавишу Ctrl выберите все необходимые таблицы. Чтобы выбрать все или блок таблиц отметьте первую таблицу, а затем удерживая клавишу Shift укажите последнюю, нажмите кнопку ***Add***. Вы также можете добавлять таблицы перетаскивая их с помощью мыши из окна базы данных в окно отношений.

Теперь с помощью мыши перетащите поля первичного ключа первичной таблицы в соответствующее положение в связываемой таблице. ***Microsoft Access*** выдаст диалоговое окно ***Relationships***.

Проверьте имена полей подлежащих связыванию, отредактируйте при необходимости. Установите соответствующие опции, например, тип отношения: ***One***(***один к одному***) или ***Many***(***один ко многим***).

Нажмите кнопку ***Create***.

Связанные таблицы, например, могут выглядеть так:

Таблицы, находящиеся в базе данных, созданной другой программой, можно импортировать в ***Access***, также можно и экспортировать ***Access-таблицы***, сохраняя их в форматах, понятных другим прикладным программам управления базами данных. Но прежде чем непосредственно приступить к импорту или экспорту, необходимо совершить соответствующие установки импорта или экспорта.

Установка опций импорта/экспорта.

В меню ***File*** выберите директиву ***Import|Export Setup***

Проще всего выбрать уже существующую спецификацию из списка **Specification Name**, и отредактировать ее при необходимости.

Если Вы будете использовать свою собственную спецификацию неоднократно, нажмите кнопку ***Save as*** и впишите имя, нажмите ***OK***.

Нажмите ***OK***.

Импортирование таблиц.

В меню ***File*** выберите директиву ***Import*** или нажмите кнопку ***Import*** в строке пиктограмм.

В диалоговом окне импорта из списка ***Data Source*** выберите нужный формат файла базы данных. Он может быть типа:

**-** Собственный формат

**-** Текст с символом - разделителем между полями

**-** Текст с фиксированным форматом полей

**-** Таблицы ***Microsoft Excel***

**-** Таблицы ***Lotus 1-2-3*** или ***1-2-3/W***

**-** Базы данных ***Paradox***

**-** Файлы ***dBASE iii*** или ***iv***

**-** ***Btrieve*** (*с файлом словаря Xtrieve*)

**-** ***Microsoft SQL -сервер***

Нажмите кнопку ***OK***. Появится диалоговое окно для выбора импортируемого файла.

Выберите нужный файл , нажмите кнопку ***Import***.

***Microsoft Access*** выдаст сообщение о результате импортирования.

Экспортирование таблиц.

В меню ***File*** выберите директиву ***Export*** или нажмите кнопку ***Export*** в строке пиктограмм.(*Первоначально нужно активизировать Базу данных из которой Вы хотите экспортировать таблицу*).

В диалоговом окне экспорта из списка ***Data Source*** выберите нужный формат файла базы данных, куда Вы хотите экспортировать таблицу. Он может быть типа:

**-** таблицы ***Microsoft Access***

**-** базы данных ***Paradox***

**-** файлы ***dBASE iii*** или ***iv***

**-** ***Btrieve*** (*с файлом словаря Xtrieve*)

**-** ***Microsoft SQL-сервер***

Нажмите кнопку ***OK***. В зависимости от выбранного формата появится соответствующее диалоговое окно для выбора таблиц из базы-источника, а затем диалоговое окно для выбора файла назначения.

Выберите соответствующие спецификации и выполните экспортирование.

***Microsoft Access*** выдаст сообщение о результате экспортирования.

**Запросы.**

Запрос можно представить себе как точку зрения на данные, включенные в таблицу. Запросы служат для селекции и фильтрации набора данных. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию(условию) и нужна для решения конкретной задачи. Например, Вам может понадобится информация о поставщиках и поставляемых ими товарах. Вы можете создать запрос типа “Какие товары поставляют московские поставщики” или “Кто из петербургских поставщиков сигарет продал за последний квартал наибольшую партию”. Результат обработки программой ***Access*** такого запроса представляет собой таблицу называемую ***Dynaset***. В эту таблицу включены выбранные из основной таблицы(или нескольких таблиц) блоки данных, которые удовлетворяют критериям запроса. ***Dynaset*** - динамический, временный набор данных, поэтому при каждом выполнении запроса он строится вновь на основе “свежих” табличных данных.

***Microsoft Access*** дает большие возможности в гибкости проектирования запросов. Вы можете:

Выбрать поля, Вам не надо включать все поля таблицы в запрос.

Выбрать записи, Вы можете специфицировать запрос таким образом что получите данные определенного вида. Например, цены на товары за определенный месяц.

Отсортировать записи, Вы можете просматривать записи в определенном порядке. Например, Вы можете вывести имена клиентов на экран в алфавитном порядке.

Запрашивать данные из нескольких таблиц, данные из нескольких таблиц могут обрабатываться вместе и Вы можете просмотреть совмещенные данные. Также Вы можете запрашивать данные из других баз данных, таких как ***Microsoft FoxPro***, ***Paradox***, ***dBASE***, ***Btrieve и*** Microsoft или ***Sybase SQL-серверы***.

Выполнять вычисления, Вы можете создавать новые поля которые будут содержать результаты вычислений.

Использовать запрос в качестве источника данных для формуляров, отчетов и других запросов. Каждый раз когда Вы будете открывать формуляр или печатать отчет, Ваш запрос будет извлекать новейшую информацию из таблиц. Вы также сможете модифицировать существующие данные прямо в формуляре основанном на запросе.

Изменять данные в таблицах, Вы можете обновлять, удалять, добавлять группы записей все сразу. Вы также можете создать новую таблицу на базе существующей таблицы или группы таблиц.

Выделяют два типа запросов:

***QBE-запросы*** (*Query by Example -Запрос по образцу*). Пользователь дает им определения, специфицируя отдельные параметры в окне проектирования с использованием подсказок(образцов).

***SQL - запросы*** (*Structured Query Language -Структурированный язык запросов* ). Пользователь формулирует их с использованием инструкций и функций, выстраивая описание. ***QBE-запрос*** ***Access*** легко транслирует в соответствующий ***SQL-запрос***. Обратная операция тоже не составляет труда. Вообще для Access безразлично, с каким типом запроса работает пользователь.

Запросы можно создавать с помощью Конструктора запросов. Он ускоряет проектирование нескольких специальных типов запросов:

Запросы кросс-таблиц. Представляют Ваши данные в компактном суммированном формате.

Запросы нахождения дубликатов. Находят дубликатные записи в выбранной Вами таблице или запросе.

Запросы нахождения несоответствия. Находят все записи в таблице которые не связаны с записями в другой таблице.

Запросы архивов. Копируют записи из существующей таблицы в новую и затем удаляют (по желанию) эти записи из таблицы оригинала.

Но для большинства общеиспользуемых запросов не нужна помощь Конструктора запросов.

Создание запроса без помощи Конструктора запросов

В окне базы данных нажмите кнопку ***Query***.

Нажмите кнопку ***New***. ***Microsoft Access*** выдаст диалоговое окно ***New Query***.

Нажмите кнопку ***New***. Откроется окно ***Select Query***.

Если Ваш запрос основывается на таблицах активизируйте индикатор ***Tables*** на панели ***View***. Если на запросах - активизируйте ***Queries***, если на обоих типах объектов выберите ***Both***. В списке ***Table / Query*** выберите необходимые для запроса источники. Для выбора более чем одной таблицы/запроса одновременно, удерживая клавишу Ctrl выберите все необходимые таблицы/запросы. Чтобы выбрать все или блок отметьте первый пункт, а затем удерживая клавишу Shift укажите последний . Нажмите кнопку ***Add***.

Нажмите кнопку ***Close***.

Окно проектирования QBE-запроса и сам QBE-запроса может выглядеть так:

Критерии отбора - это инструкции, посредством которых пользователь сообщает ***Access*** ,какие блоки данных должны отбираться по запросу и индицироваться в ***Dynaset***. Критерии могут задаваться для одного или нескольких полей запроса.

Спецификация критерия запроса отбора.

Войдите в режим проектирования запроса, для этого из меню ***View*** выберите директиву ***Query Design*** или щелкните мышью на кнопке ***Design View*** в пиктографическом меню.

В нижней части окна проектирования, в строке ***Criteria*** в ячейке ***Category Name*** введите выражение критерия отбора.

После того как выражение введено и нажата клавиша ввода, ***Access*** проверяет корректность выражения по правилам записи критериев и нормализует запись по этим правилам.

Пользователь может воспользоваться запросом для проведения вычислений с блоками данных. Он может задать в каждом поле некоторую функцию, обрабатывающую содержимое этого поля. Результат обработки выдается в ***Dynaset***. Функция обработки задается в строке ***Total***, которая появляется после нажатия в пиктографическом меню кнопки с греческой литерой “*сигма*”. Саму функцию можно выбрать в этой строке, развернув список возможных значений.

После того как запрос спроектирован, его можно выполнить, щелкнув мышью в строке пиктограмм на кнопке с восклицательным знаком или вызывая в меню ***Query*** директиву ***Run***. ***Access*** индицирует отобранные по запросу в ***Dynaset*** блоки в виде таблицы.

С помощью структурированного языка запросов ***SQL*** в рамках ***Access*** пользователь может сформулировать сколь угодно сложные по структуре критериев и вычислений запросы. Этот же язык позволяет управлять обработкой запросов. ***SQL-запрос*** представляет собой последовательность инструкций, в которую могут включаться выражения и вызовы агрегатных функций. Начинающим пользователям рекомендуется создавать запросы в режиме проектирования, выбирая необходимые поля, критерии сортировки и функции в зоне спецификации. Для описания запросов на языке ***SQL*** нужен достаточный опыт работы с базами данных не в качестве пользователя, а в качестве программиста.

Если пользователь хочет иметь разработанный проект запроса в своем распоряжении и во время следующих сеансов работы, то он должен его сохранить с помощью директивы ***Save*** в меню ***File***. Если проект сохраняется впервые, то после обращения к этой директиве появится диалоговое окно ***Save As***. В нем следует назначить запросу имя, под которым он будет сохранен и включен в список запросов в окне банка данных. Имя запроса не должно совпадать с именем таблицы.

Если данный запрос уже был однажды сохранен и пользователь выбрал директиву ***Save***, то старая версия будет заменена новым, измененным проектом запроса. Если изменения не должны испортить предыдущую версию, а должны быть сохранены в другом запросе, то следует вызвать директиву ***Save As*** и назначить сохраняемому проекту новое имя. Для сохранения табличного представления запроса следует выбрать директиву ***Save Query*** в меню ***File***.Эта директива сохраняет изменения в проекте имеющегося запроса и заменяет предыдущую версию запроса на активную версию. Для того чтобы сохранить запрос впервые или создать копию активного запроса, ***Dynaset*** которого виден на экране, следует выбрать директиву ***Save Query As***.

Наряду с запросами выбора, с помощью ***Access*** можно реализовать также запросы действий, параметрические запросы и запросы кросс-таблиц.

**Запросы действия**.

С помощью запроса действия пользователь может изменять или переносить данные таблицы, а также актуализировать, добавлять или удалять группы блоков данных ,изготавливать новые таблицы из набора ***Dynaset***. Различают четыре типа запросов действий: запрос добавления, запрос удаления, запрос актуализации и запрос создания таблицы.

**Параметрические апросы.**

Часто используются запросы,которые представляют собой незначительно

видоизмененные варианты однажды подготовленного базового запроса. Речь может идти, например, о запросе, который позволяет выбирать из базы данных телефоны клиентов определенного региона, причем сам регион задается отдельно, в диалоге. Такие запросы, параметрические, видоизменяются от случая к случаю, но незначительно. Для их реализации проектируется один параметрический запрос, в котором указывается тот критерий (критерии), который может изменяться по заказу пользователя.

**Запросы кросс-таблиц.**

Если необходимо объединить данные в формате строк-колонок (двумерная таблица), то следует изготовить запрос кросс-таблицы. При проектировании запроса кросс-таблицы можно указать в качестве заголовков для колонок значения некоторых полей или выражений. Запросы кросс-таблиц позволяют более компактно, чем обычные запросы, индицировать данные, объединяя однотипную информацию.

**Формуляры.**

Просмотр базы данных в виде таблицы в режиме заполнения дает пользователю возможность оценить базу как единое целое, сравнить записи и т.п. Часто, однако, возникает необходимость работы с отдельными записями базы. В этом случае присутствие на экране других записей (как это имеет место в режиме заполнения) только мешает и отвлекает. Работа с отдельными записями посредством формуляров позволяет сосредоточиться только на относящейся к делу информации.

Формуляр представляет собой бланк, подлежащий заполнению, или маску, накладываемую на набор данных. Бланк-формуляр позволяет упростить процесс заполнения базы данных, благодаря чему появляется возможность поручить ввод информации персоналу невысокой квалификации. Маска-формуляр позволяет ограничить объем информации, доступной пользователю, обращающемуся к базе. Речь здесь также может идти о блокировании индикации служебных или засекреченных полей.

Проще всего создать формуляр спомощью Конструктора.

окне базы данных активизируйте кнопку ***Form***. При этом ***Microsoft Access*** покажет набор существующих формуляров Вашей базы данных, если они есть.

Нажмите кнопку ***New***. ***Microsoft Access*** высветит диалоговое окно ***New Form***.

строке ***Select A Table/Query*** выберите необходимую таблицу или запрос.

Нажмите кнопку ***Form Wizard. Microsoft Access*** высветит первое диалоговое окно ***Form Wizard***.

Выберите ***AutoForm***. Нажмите ***OK***.

В других окнах ***Form Wizard*** Вам будут предложены вопросы, и на основе Ваших ответов Конструктор создаст формуляр.

Если пользователь уверен в том, что сможет самостоятельно спроектировать формуляр без помощи Конструктора Формуляров (вполне реальная ситуация при наличии достаточного опыта), то ему следует создать пустой формуляр и затем внести в него требуемые поля, элементы оформления и управляющие элементы.

окне базы данных активизируйте кнопку **Form**. При этом **Microsoft Access** покажет набор существующих формуляров Вашей базы данных, если они есть.

Нажмите кнопку **New**. **Microsoft Access** высветит диалоговое окно **New Form**.

строке **Select A Table/Query** выберите необходимую таблицу или запрос.

Нажмите кнопку **Blank Form** (Пустой формуляр). **Access** индицирует окно формуляра в режиме проектирования.В нем можно разместить поля и другие элементы.

Внешний вид, структура и режимы работы отдельных управляющих элементов и формуляров определяются значениями характеристик этих объектов. Характеристики объектов доступны пользователю через окно характеристик. Для установки или изменения характеристик элемента его надлежит маркировать и из меню ***View*** вызвать директиву ***Properties***. Можно также выполнить двойной щелчок мышью на том элементе, окно свойств которого необходимо вывести на экран.

Для индикации инструментального окна установите в меню ***View*** опцию ***Toolbox***. Каждый инструмент, а точнее, создаваемый с его помощью элемент управления/оформления, располагает собственным окном характеристик, и в нем в виде списка представлены стандартные установки для соответствующего управляющего элемента.

Включаемые в формуляр поля можно выбрать из списка полей, который будет виден на экране после обращения к директиве ***Field List*** в меню***View***.

При проектировании формуляра в него можно вставить управляющие элементы. Речь идет, например, о командных кнопках, списках, иллюстрациях и пр. Каждый управляющий элемент может быть связан с некоторым объектом (полем) в таблице (или запросе), на которой базируется форма.Например, связанное с некоторым полем блока таблицы текстовое поле формуляра будет постоянно отображать содержимое “своего” поля активной записи таблицы. Выбранные и размещенные в окне проектирования формуляра элементы управления приобретают свой реальный вид только в режиме заполнения или при просмотре прототипа распечатки (режим ***Preview***). В режиме же проектирования они обычно представлены условными обозначениями, зачастую близкими к реальному виду обозначаемых элементов.

Для размещения в формуляре несвязанного управляющего элемента следует вызвать на экран инструментальное меню (***View/Toolbox***), щелкнув в нем мышью на кнопке с изображением размещаемого элемента и разместить элемент в окне проектирования фиксацией курсора мыши на нужном месте в прототипе формуляра. В результате таких действий в прототип формуляра будет помещен выбранный контрольный элемент с характеристиками по умолчанию. Для создания управляющего элемента “нестандартного” размера после фиксации курсора мыши следует, не отпуская кнопку мыши, переместить ее, обозначив тем самым главную диагональ прямоугольника, охватывающего размещаемый контрольный элемент.

|  |  |  |
| --- | --- | --- |
| Указатель  Надпись  Группа  Селекторная кнопка  Комбинированный список  Диаграмма  Рамка несвязанного объекта  Линия  Конец страницы    Конструктор управления |  | Текстовое поле  Двухпозиционный переключатель  Контрольный индикатор  Список  Подформа/подотчет  Рамка связанного объекта  Прямоугольник  Командная кнопка  Блокировка |

Существует два способа связывания контрольного элемента с полем таблицы:

Если пользователь хочет, чтобы ***Access*** изготовила связанный управляющий элемент автоматически, то при генерации и размещении элемента кроме инструментального меню (***View/Toolbox***) ему следует воспользоваться еще и окном списка полей (***View/Field List***). Из инструментального окна мышью выбирается управляющий элемент, который необходимо вставить в формуляр. Затем из списка полей мышью выбирается поле, с которым следует связать создаваемый управляющий элемент.После этого, не отпуская кнопку мыши, следует отбуксировать выбранное поле в то место прототипа формуляра, в котором надлежит разместить управляющий элемент. При освобождении кнопки мыши в указанном месте создается элемент выбранного типа, привязанный к указанному полю. Такой управляющий элемент имеет те же характеристики, что и поле в таблице, скоторым он связан.

Связать с полем таблицы можно и ранее не связанный управляющий элемент. Для этого достаточно вызвать окно характеристик управляющего элемента двойным щелчком на этом элементе в окне проектирования формуляра. В окне характеристик следует найти характиристику ***ControlSource*** (*Источник управления*) и задать в качестве ее значения имя связанного поля.

Если Вы бы хотели распечатать Ваш формуляр, то сначала рекомендуется просмотреть его в виде, в котором он будет представлен на странице.

Для этого:

Нажмите кнопку ***Print*** ***Preview*** в пиктографическом меню (или выберите директиву ***Print Preview*** в меню ***File***).

Для увеличения масштаба установите указатель мыши на странице так чтобы он приобрел вид увеличительного стекла и нажмите кнопку мыши.

Для возвращения прежнего масштаба щелкните кнопкой мыши еще раз.

Для распечатки формуляра необходимо:

Из меню ***File*** выбрать директиву ***Print*** (*или нажать кнопку Print в пиктографическом меню*). ***Microsoft Access*** высветит диалоговое окно ***Print***.

Нажмите ***OK***. ***Microsoft Access*** распечатает формуляр.

Теперь из меню ***View*** выберите директиву ***Form*** (*или нажмите кнопку Close в пиктографическом меню*) для возвращения в режим просмотра формуляра.

Теперь Вы можете сохранить и закрыть формуляр.

Сохранение:

Из меню ***File*** выбрать директиву ***Save Form***. ***Microsoft Access*** высветит диалоговое окно ***Save As***.

Напечатайте имя Вашего формуляра и нажмите ***OK***.

Закрытие:

Из меню ***File*** выбрать директиву ***Close*** (или сделать двойной щелчок в поле меню управления в вержнем левом углу окна формуляра.

Теперь формуляр стал частью Вашей базы данных.

**Отчеты.**

**Отчет** это информация, которую Вы оформили в соответствии с Вашими спецификациями. Отчет позволяет извлекать и представлять данные как значимую информацию, которую Вы можете использовать и распространять. Примерами отчетов могут служить почтовые адреса, накладные, суммы продаж или списки телефонов. С помощью ***Microsoft Access*** Вы можете спроектировать отчет который представит информацию в том виде в котором Вы пожелаете. Вы можете использовать множество различных элементов проектирования, таких как текст, данные, рисунки, линии, поля и графики для создания Вашего отчета.От Вас зависит, как и какие элементы использовать для построения отчета.

***Microsoft Access*** обеспечивает несколько способов получения информации из Вашей базы данных - используя запрос, формуляр или отчет.Вы можете выбрать тот метод, который наиболее подходит к решению Вашей задачи.

Например:

Для просмотра всех товаров определенной торговой поставки, используйте запрос.

Для просмотра полной информации только об одном товаре используйте формуляр.

Для распечатки торговых продаж для формального прелставления, используйте отчет.

Отчеты также являются эффективным средством для распечатки информации регулярного использования. Вы можете создать проект отчета и сохранить его для дальнейшего использования. Проект отчета останется неизменным, но каждый раз Вы будете получать распечатку текущих данных.

Перед тем как создать отчет, подумайте о том откуда Ваш отчет будет извлекать данные. Если все данные из одной таблицы, то Ваш отчет будет основываться на этой таблице.Если же Вам необходимы данные более чем из одной таблицы, то Ваш отчет должен основываться на запросе.

Вы можете создавать отчеты как при помощи так и без помощи Конструктора Отчетов. Конструктор намного ускорит процесс создания отчета, так как он сделает за Вас основную работу.

Самый наибыстрый способ создания отчета - используя Конструктор Автоотчета.Отчет будет представлять собой единственную колонку, показывая все поля таблицы или запроса.

С помощью Конструктор Автоотчета:

В окне базы данных активизируйте ***Table*** или ***Query***.

Выберите таблицу или запрос на котором будет основываться Ваш отчет.

Нажмите кнопку ***AutoReport*** в пиктографическом меню.

С помощью Конструктора Отчета:

В окне базы данных активизируйте ***Report*** (*или выберите директиву Reports в меню View*).

Нажмите кнопку ***New*** (*или выберите директиву New в меню File, а затем активизируйте Report*). ***Microsoft Access*** высветит диалоговое окно ***New Report***.

В строке ***Select A Table/Query*** выберите таблицу или запрос, содержащий данные для Вашего отчета.

Нажмите кнопку ***Report Wizard***. ***Microsoft Access*** высветит первое диалоговое окно ***Report Wizard***.

Выберите Конструктор из списка и нажмите ***OK***.

Следуйте указаниям в диалоговых окнах Конструктора. В последнем окне нажмите кнопку ***Finish***.

Без помощи Конструктора Отчета:

В окне базы данных активизируйте ***Report*** (*или выберите директиву Reports в меню View*).

Нажмите кнопку ***New*** (*или выберите директиву New в меню File, а затем активизируйте Report*). ***Microsoft Access*** высветит диалоговое окно ***New Report***.

В строке ***Select A Table/Query*** выберите таблицу или запрос, содержащий данные для Вашего отчета.

Нажмите кнопку ***Blank Report***.

***Microsoft Access*** высветит отчет в режиме проектирования. Для просмотра списка полей таблицы или запроса нажмите кнопку ***Field List*** в пиктографическом меню.

В прототип отчета в процессе проектирования можно встроить управляющие элементы. Управляющие элементы, встроенные в прототип отчета, могут использоваться для оформления заголовков, надписей и иллюстраций.

Для встраивания управляющих элементов в отчет на экране должно присутствовать инструментальное меню ***Toolbox*** (*опция View/Toolbox*). В инструментальном меню выбирается кнопка создаваемого управляющего элемента. После этого следует выполнить мышью щелчок в той позиции прототипа отчета, где надлежит разместить данный управляющий элемент. ***Access*** размещает выбранный элемент, приписывая ему по умолчанию стандартные значения характеристик (размер, цвет и т.п.).

|  |  |  |
| --- | --- | --- |
| Указатель  Надпись  Группа  Селекторная кнопка  Комбинированный список  Диаграмма  Рамка несвязанного объекта  Линия  Конец страницы    Конструктор управления |  | Текстовое поле  Двухпозиционный переключатель  Контрольный индикатор  Список  Подформа/подотчет  Рамка связанного объекта  Прямоугольник  Командная кнопка  Блокировка |

Для изменения стандартных значений характеристик управляющего элемента можно обратиться к окну характеристик, предварительно маркировав настраиваемый элемент. Окно характеристик видно на экране при установленной опции ***View/Properties***. Размер элемента и его местоположение можно изменить “прямо на месте”, перемещая с помощью мыши весь элемент или его размерные манипуляторы.

Зависимый управляющий элемент в отчете связан с полем в соответствующей таблице. С помощью зависимого (связанного) управляющего элемента можно ввести данные в поле или индицировать и актуализировать их в отчете.

Связывание управляющего элемента можно выполнить одним из двух способов:

Если пользователь хочет, чтобы Access автоматически создала управляющий элемент, связанный с базовой для данного отчета таблицей, то следует воспользоваться списком полей таблицы. Он виден на экране, если установлена опция ***View/Field*** ***List***. правляющий элемент, изготовленный с помощью списка полей, имеет те же значения характеристик, что и поле таблицы, с которым он связан. Для выполнения связывания следует отбуксировать выбранное поле или поля из списка в отчет. Там надлежит поместить курсор мыши в позицию, где должен находиться левый верхний угол управляющего элемента (но не поля названного данного управляющего элемента), и затем отпустить кнопку мыши.Access создает для каждого поля, выбранного в списке полей, связанный управляющий элемент (текстовое поле).

Если задним числом необходимо связать с полем уже имеющийся, несвязанный управляющий элемент, то это можно выполнить с помощью окна характеристик.В этом окне после маркирования управляющего элемента в прототипе отчета перечислены характеристики этого элемента. При занесении имени связываемого поля в это окно в качестве значения характеристики ***ControlSource*** выполняется связывание.

Вы можете распечатать отчет посылая его прямо на принтер или пересылая его в файл, который Вы распечатаете позже.

Печать на принтер:

Откройте отчет в любом режиме.

Из меню ***File*** выберите директиву ***Print*** . ***Microsoft Access*** выдаст диалоговое окно ***Print***.

Выберите опции:

-***All*** распечатка полного отчета

-***Selection*** распечатка выборочной области

-***Pages*** распечатка диапазона страниц, который Вы определяете в стоках ***From*** и ***To***.

В строке ***Print Quality*** (*качество печати*) выберите соответствующую опцию.Они будут зависить от драйвера принтера.

Для установки числа копий введите номер в строку ***Copies***.

После установки всех необходимых опций нажмите ***OK***.

Если принтер Вам недоступен Вы можете переслать отчет в файл.

Откройте отчет в любом режиме.

Из меню ***File*** выберите директиву ***Print*** . ***Microsoft Access*** выдаст диалоговое окно ***Print***.

Выберите опции:

-***All*** распечатка полного отчета

-***Selection*** распечатка выборочной области

-***Pages*** распечатка диапазона страниц, который Вы определяете в стоках ***From*** и ***To***.

Активизируйте строку ***Print To File***.

Нажмите ***OK***. Microsoft Access выдаст диалоговое окно Print To File.

Напечатайте имя файла назначения и нажмите ***OK***.

Вы можете сохранить отчет двумя способами:

Сохраняя только его проект, структуру отчета, без данных.

В меню ***File*** выберите директиву ***Save***.

Если отчет не был назван, напечатайте имя в строке ***Report Name***.

Нажмите ***OK***.

Сохраняя результат, полный отчет - проект вместе с данными - в файле, который Вы сможете использовать в других поддерживаемых ***Windows*** приложениях, например, ***Microsoft Excel***.

В меню ***File*** выберите ***Output To***.

Следуйте инструкциям в диалоговых полях ***Output To***.

**Макросы, макрокоманды.**

**Макрокоманда** представляет из себя последовательность операций, записанных в виде инструкций на специальном языке. При оработке данных часто приходится делать длинные последовательности действий. Но любые действия, операции могут быть записаны в виде макрокоманд. Вызов макрокоманд приводит к выполнению последовательности ранее описанных действий. Применение макрокоманд позволяет автоматизировать различные повторяющиеся процессы, такие как заполнение Базы, выполнение запросов, открывание различных объектов и т.д.

Последовательность операций вводится пользователем в окно проектирования макокоманд. Для создания макрокоманды в окне Базы данных нажмите кнопку ***Macro***, а затем кнопку ***New***. После этого откроется окно проектирования макрокоманд. Полный вид этого окна:

В окне проектирования макрокоманд каждая строка может содержать одну инструкцию в колонке ***Action*** и комментарий к ней в колонке ***Comment***. Нужную инструкцию вы можете выбрать из предоставляемого списка.

Колонка ***Condition*** содержит условные выражения для управления процессом выполнения макрокоманды. Эта колонка не обязательна и при первоначальной загрузке окна проектирования макрокоманд она не высвечивается. Выбрать ее Вы можете в меню ***View*** выбрав опцию ***Conditions***.

Вы так же можете создать макрогруппу содержащую несколько макрокоманд. Каждая из макрокоманд будет выполнятся независимо от других макрокоманд в группе. Макрокоманды в ходящие в макрогруппу представляют собой именованные последовательности инструкций. Создать макрогруппу Вы можете так: сначала создайте макрокоманду, а затем в колонке ***Macro Name*** в первой строке задайте ей имя. Эта колонка так же не обязательна и при первоначальной загрузке окна проектирования макрокоманд она не высвечивается. Выбрать ее Вы можете в меню ***View*** выбрав опцию ***Macro Names***.

В одном окне проектирования макрокоманд может содержаться несколько макрогрупп.

Выполнить макрокоманду Вы можете нажав на кнопку

или выбрыв опцию ***Run Macro...*** в меню ***File***.

**Модули.**

При решении достаточно сложных задач Вы скоро столкнетесь с ограниченностью макрокоманд. Для преодоления возникших проблем Вы можете попробовать написать свою программу на встроенном в ***Access*** языке программирования - ***Access Basic***. Он представляет из себя средство визуального программирования и является диалектом языка **Visual Basic**. Такая процедура оформляется как **Модуль**.

**Модуль** представляет из себя совокупность объявлений (деклараций) и последовательностей исполняемых команд (процедур, функций) записанных на язаке ***Access Basic***.

Для создания модуля нажмите на кнопку ***Modul*** в окне Базы данных, а затем на кнопке ***New***. Вы увидете такое окно:

Это окно представляет из себя текстовый редактор, с автоматической проверкой синтаксиса при заканчивании строки, где вы можете набирать текст своей программы состояший из переменных, операторов, вызываемых функций.

***Access Basic-код***(*последовательность Ваших операторов*) выполняется при вызове и выполнии операции в которой используется Ваша функция, при нажатии кнопки:

**3.** **ПОСТРОЕНИЕ НЕБОЛЬШОЙ БАЗЫ ДАННЫХ.**

Для того чтобы лучше понять ***Access*** построим свою Базу данных. Сначала определимся какую информацию будет содержать наша База данных. База данных может содержать информацию о чем угодно: о зарплате, о торговых поставках, о больных в поликлинике, о студентах ВУЗа, о друзьях, о книгах в вашей библиотеке. Мы постоим самую простую - о студентах какого-нибудь ВУЗа. Эта База данных будет состоять из трех взаимосвязанных таблиц:

Список ВУЗов, который будут содержать такую информацию:

**-** Номер ВУЗа.

**-** Название ВУЗа.

Список факультетов, который будут содержать такую информацию:

**-** Номер факультета.

**-** Номер ВУЗа, где находится этот факультет.

**-** Название факультета.

Список студентов, который будут содержать такую информацию:

**-** Номер студента.

**-** Номер факультета.

**-** студента.

**-** На каком курсе студент.

И это далеко не полный список того что мы можем знать о студенте, но ведь мы делаем небольшую Базу и этого хватит что бы понять как строить Базу данных.

Графически это может быть представлено в виде трех таблиц:

1. Список ВУЗов:

|  |  |
| --- | --- |
| Номер ВУЗа | Название ВУЗа |
| 1 | ПВУ |
| 2 | ППИ СПбГТУ |
| 3 | ПГПИ |

2. Список факультетов:

|  |  |  |
| --- | --- | --- |
| Номер факультета | Номер ВУЗа | Название факультета |
| 1 | 1 | Информационные системы |
| 2 | 3 | Ин. яз. |
| 3 | 2 | Мех. Маш. |

3. Список студентов:

|  |  |  |  |
| --- | --- | --- | --- |
| Нмер студента | Номер факультета | Ф.И.О. | Курс |
| 1 | 3 | Иванов И.И. | 5 |
| 2 | 2 | Сидоров С.С. | 1 |
| 3 | 1 | Петров П.П. | 1 |

Где стрелочками показаны связи или отношения(реляции) между таблицами (*т.к. Access реляционная СУБД*).

Итак, начнем.

Запускаем ***Access*** и открываем Базу данных, назовем ее ***Students***. Сначала нам надо построить три, выше описанных таблицы. Для этого нажимаем кнопку ***New*** и начнем конструировать первую таблицу. Вводим имена поле и их типы, первой записью вы можете ввести счетчик(*тип counter*) записей и установить первичный ключ, но это необязательно. ***Access*** при сохранении таблицы спросит: "Создать первичный ключ?" если Вы выберите Да, то он создаст новое поле с именем ***ID*** и типом ***counter***. Дальше нам будут нужны только поля с типами ***Text*** и ***Number***.

Эти таблицы будут выглядеть так:

1.

2.

3.

Дальше нам нужно установить связи между таблицами, для этого в меню ***Edit*** выберите опцию ***Relationships*** и установите связи между полями. Для установления связи выберите нужное вам поле, укажите на него мышкой и удерживая правую кнопку мыши покажите на поле с которым Вы хотите установить связь. Но здесь есть небольшое ограничение: поля между которыми Вы хотите установить связь должны быть одного типа. После установления связей наши таблицы будут выглядеть так (в данном случае мы установили связь один ко многим):

Эти связи нам нужны для облегчения поиска и выборки информации. Например при просмотре списка университетов ***Access*** автоматически устанавливает указатель(т.е. делает запись текущей) на первую запись в таблице ***Faculties*** содержащую в поле ***Namber\_of\_univer*** такое же число, что и в текущем поле ***ID*** таблицы ***Univer***.

Теперь мы можем заполнить какой-либо информацией эти три таблицы. Для этого в окне Базы Данных нажмите кнопку ***Open*** или в меню ***View*** выберите опцию ***Datasheet***. И начинайте заполнять, поля с типом ***Cоunter*** заполняются автоматически. После заполнения, содержимое этих таблиц может выглядеть, например, так:

1.

2.

3.

Как вы понимаете это минимальный набор информации по этой теме, но нам и его хватит.

И так на данном этапе мы имеем Базу данных состоящую из трех связанных таблиц и содержащую некоторую информацию. Дальше нам надо как-то работать с этой Базой и в первую очередь нам надо производить поиск и выборку информации. Для этого существуют запросы.

Построим несколько запросов.

1. Отсортируем таблицу student в алфавитном порядке по полю name. Для этого сделаем ***QBE-запрос*** такого вида:

И если мы просмотрим ***Dynaset*** то увидим следующую таблицу:

2. Дальше нам, например, может понадобиться выбрать из списка студентов всех обучающихся на первом курсе. Для этого существуют критерии выборки. Для того чтобы осуществить нужную нам выборку нам надо в ***QBE-запросе*** в поле ***Criteria*** столбца ***year*** поставить цифру ‘1’(*это значит что из таблицы Student будут выбраны все записи у которых в поле столбца year стоит ‘1’*). И если мы просмотрим ***Dynaset*** то увидим следующую таблицу:

3. Так как у нас есть несколько связанных таблиц, то мы можем сделать запрос основанный на нескольких таблицах. Например, из имеющихся у нас таблиц сделаем такую выборку: Название ВУЗа, название факультета где обучается студент, имя студента и курс, отсортированную по полям ***name\_of\_university*** и ***name***. Для этого подготовим такой ***QBE-запрос***:

И если мы просмотрим ***Dynaset*** то увидим следующую таблицу:

Таким образом вы видите что ***Access*** позволяет осуществлять различную обработку информации.

Но что Вы будете делать если в Вашу Базу данных ввод информации будет осуществляется неквалифицированным пользователем. Чтобы такие люди могли спокойно работать с вашей Базой и не видели того, что им ненужно видеть, ***Access*** предусматривает такую вещь как *Формуляры*. Построим один.

Из предусмотренных ***Access*** способов создания формуляров выберем самый простой - это использование Конструктора формуляров. Свой формуляр сделаем на основе запроса который мы, в свою очередь сделали на основе нескольких таблиц. Из предлагаемых формуляров выберем ***Single-column***, дальше выберем все поля и сделаем его ***Shadowed***. После всего этого у нас должно получиться что-то вроде этого:

В этом окне можно как редактировать так и вводить новую информацию. Помимо этого, формуляры можно распечатывать на принтере. Посмотреть как будет выглядеть данный формуляр на листе бумаге Вы можете в конце работы.

Кроме формуляров ***Access*** предоставляет специальное средство для распечатки выбранной информации - *Отчеты*. Отчеты отличаются от формуляров одним: в отчете нельзя редактировать данные. Давайте создадим один, но основанный только на одной таблице ***Univer***. Для этого также воспользуемся Конструктором отчетов, т.к. это более легкий и понятный способ построения чего-либо. Для этого из предлагаемых запросов и таблиц выберем таблицу ***Univer***, потом выберем опцию группировки ***Groups/Totals***, порядок полей в отчете оставим такой же, сгруппируем по полю ***name\_of\_university*** в режиме ***Normal***, очередность сортировки в группах сделаем по оставшемуся полу ***ID***, выберем стиль формы отчета ***Presentation*** и расположение на листе бумаги ***Portrait***, введем имя для отчета ***University*** и нажмем кнопку ***Finish***. ***Access*** создаст отчет, его Вы сможете увидеть в конце работы.

Обрабатывая таким образом информацию мы часто повторяем определенные последовательности операций и для того чтобы избежать этого, а также повысить эффективность общения с Базой данных и сократить время обработки данных. **Access** предоставляет набор *Макрокоманд*. В нашей Базе есть одно такое повторяющееся действие: если при каждой загрузке Базы данных мы хотим открывать сразу все наши таблицы, запросы, отчеты или что-нибудь еще, то мы можем записать последовательность макрокоманд делающих это.

Для этого создадим окно проектирования макрокоманд, здесь каждая строка может содержать одну инструкцию в колонке ***Action*** и комментарий к ней в колонке ***Comment***. В каждой строке колонки ***Action*** есть список макрокоманд из которого выберем нужные нам макрокоманды: ***Opentable***(*Открыть таблицу*), ***OpenQuery***(*Открыть запрос*), ***OpenReport***(*Открыть отчет*), ***Minimize***(***Минимизировать***), ***MsgBox***(*Окошко сообщения*) и разобьем их на макрогруппы с именами 1, 2, 3. Первая макрогруппа будет выдавать сообщение, открывать три таблицы и минимизировать их. Вторая будет выдавать сообщение, открывать запрос и минимизировать его. Третья будет выдавать сообщение, открывать отчет и минимизировать его.

Окно проектирования макрокоманд будет выглядеть так:

Теперь чтобы запустить это, надо в меню ***File*** выбрать опцию ***Run Macro*** и выбрать из списка нужную вам макрогруппу. Например, если вы запустите первую, то сначала увидите сообщение:

Если Вы нажмете на кнопку OK, то увидите что-то вроде этого:

Наша макрогруппа отработала: открыла таблицы и минимизировала их. Если вы запустите остальные, то результат будет такой же, только открыты будут запрос и отчет.

Работая с макрокомандами Вы скоро увидете, что это довольно ограниченный инструмент управления Базой данных. Помимо макрокоманд Access имеет втроенный язык програмирования - Access Basic, этот язык представляет собой диалект языка Visual Basic . Для написания программ Access предоставляет такое средство как Модуль. Модуль представляет собой небольшой текстовый редактор с проверкой ситаксиса при завершении строки. Здесь Вы можете писать свои функции и процедуры. Access Basic, как и любой язык програмирования, имеет переменные, константы, операторы цикла, условия и набор функций для выполнения различных действий.

**4. Небольшая сравнительная характеристика СУБД Clipper и Access.**

Проведем небольшие сравнение этих двух СУБД. Их главное различие это то что Clipper предназначен для работы в DOS, а Access - в Windows. Существует также различие в структуре обрабатываемых данных. Access, как было сказано выше, поддерживает, без конвертирования, такие форматы как ***dBASE***, ***Paradox*** или ***Btrieve.*** Clipper в этом отношении намного беднее, он поддерживает только dDASE формат.

***Clipper*** отличается более развитым языком программирования в этом отношении он больше похож на Си, Паскаль, чем на чисто СУБД, такую как Access, например. Конечно в ***Access*** тоже входит язык программирования ***Access Basic*** (*встроенный диалект языка Visual Basic* ), дающий возможность формирования специализированных систем управления базами данных, но более бедный.

***Access*** ориетирован больше на людей не занимаюшихся программированием. Он имеет очень удобный и понятный интерфейс, очень многие операции у него автоматизированы с помощью различных Конструкторов. Так же не надо придумывать и писать свой интерфейс для своей Базы данных, надо только выбрать из уже сушествующих. ***Clipper*** в этом отношении намного беднней и не имеет таких средст автоматизации проектирования и обработки данных.

На этом основные различия заканчиваются. Так как ***Access*** и ***Clipper*** являются реляционными СУБД, то основные термины практически не отличаются. И там, и там пользователь имеет дело с таблицами, полями, записями.

**Список литературы:**

**Microsoft Access, User’Guide. Microsoft Corporation, 1994.**

**Microsoft Access, Getting Started. Microsoft Corporation, 1994.**

**Бемер С., Фратер Г.: Microsoft Access для пользователя. Микап, Москва 1994.**