**Тема: Базы данных. Основные понятия.**

**Создание и редактирование структуры таблицы БД.**

**Цели:**

1. Формирование представления об информационных системах.

2. Формирование представления об основных различиях информационных систем от баз данных.

3. Знакомство с основными принципами проектирования БД.

4. Знакомство с интерфейсом Microsoft Access и его основными компонентами.

5. Обучение способам создания баз данных (с помощью мастера, конструктора).

6. Развитие логического мышления.

7. Развитие самостоятельности.

**Ход урока**

**I этап: Актуализация знаний**.

- Сегодня мы приступаем с вами к изучению новой темы, нового приложения пакета Office. В течении нескольких уроков мы познакомимся с такими новыми понятиями, как «информационные системы», «базы данных», «системы управления базами данных»…Узнаем интерфейс программы, в которой в дальнейшем будем работать, а также ее основные компоненты.

**II этап: Теоретическая основа урока.**

Любой из нас, начиная с раннего детства, многократно сталкивался с "базами данных". Это – всевозможные справочники, энциклопедии ... Записная книжка – это тоже "база данных", которая есть у каждого из нас.

**В широком смысле база данных (БД) — совокупность определенным образом организованной информации на какую-то тему.**

**Примеры:** база данных книжного фонда библиотеки; база данных учащихся школы; база данных законодательных актов в области уголовного права; база данных современных песен...

БД

Документальные

Фактографические

**В фактографических БД с**одержатся краткие сведения об описываемых объектах, представленные в строго определенном формате.

Из приведенных выше примеров две первые БД, скорее всего, будут организованы как фактографические. В БД библиотеки хранятся библиографические сведения о каждой книге: год издания, автор, название и пр. Разумеется, текст книги в ней содержаться не будет. В БД учащихся школы хранятся анкетные данные об учениках: фамилия, имя, отчество, год и место рождения и т. д.

**Базы данных в третьем и четвертом примерах наверняка** будут организованы как документальные. Первая из них будет включать в себя тексты законов; вторая — тексты и ноты песен; биографическую и творческую справочную информацию о композиторах, поэтах, исполнителях; звуковые записи и видеоклипы. **Следовательно, документальная БД содержит обширную информацию самого разного типа: текстовую, графическую, звуковую.**

Современные информационные технологи стирают границу между фактографическими и документальными БД. Существуют средства, позволяющий легко подключать любой документ (текстовый, графический, звуковой) к фактографической базе данных**. Далее, в основном, будем вести речь о фактографических базах данных**. Конечно, вся эта информация может храниться и на бумаге (например, книжный каталог библиотеки). Но современным средством хранения и обработки баз данных является, безусловно, компьютер. В дальнейшем мы **будем иметь в виду только компьютерные БД.**

**В информатике базой данных называют информационную модель позволяющую упорядочивать и хранить данные о группе объектов, обладающих одинаковым набором свойств.**

БД

Иерархические графически могут быть представлены как дерево, состоящее из объектов различных уровней. Верхний уровень занимает один объект, второй — объекты второго уровня и т. д.

Реляционные (табличные)

В реляционных базах данных все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Запросы к таким таблицам возвращают таблицы, которые сами могут становиться предметом дальнейших запросов. Каждая база данных может включать несколько таблиц.

Если различные части одной базы данных хранятся на множестве компьютеров, объединенных между собой сетью, то такая БД называется распределенной базой данных.

Очевидно, информацию в Интернете, объединенную «паутиной» WWW, можно рассматривать как распределенную базу данных. Распределенные базы данных создаются так же и в локальных сетях.

Сетевая база данных образуется обобщением иерархической за счет допущения объектов, имеющих более одного предка, т. е. каждый элемент вышестоящего уровня может быть связан одновременно с любыми элементами следующего уровня. Вообще, на связи между объектами в сетевых моделях не накладывается никаких ограничений. (например всемирная паутина глобальной сети)

**В настоящее время наибольшее распространение получили**

**реляционные базы данных.**

Главное достоинство таблиц — в их понятности.

С табличной информацией мы имеем дело практически каждый день.

Загляните, например, в свой дневник: расписание занятий там представлено

в виде таблицы. Когда мы приходим на вокзал, смотрим расписание

электричек. Какой вид оно имеет? Это таблица!

А еще есть таблица футбольного чемпионата.

И журнал учителя, куда он выставляет вам оценки, — тоже таблица.

Видите, как много примеров, и их еще можно продолжить. Мы настолько привыкли к таблицам, что обычно не требуется никому объяснять, как ими пользоваться. Ну разве что маленькому ребенку, который только учится читать.

**Термины БД и СУБД чаще всего употребляют как относящиеся к компьютерам.**

**СУБД** - -программы, управляющие обработкой и хранением данных. (# Microsoft Access **(.mdb) ).**

Понятие БД можно применить к любой связанной между собой по определенному признаку информации, хранимой и организованной особым образом - как правило, в виде таблицы.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип процессора** | **ОП** | **Записи** |
| Compag  Dell  IBM | Celeron  PentiumII  Pentium IV | 64  128  256 |
| **Поля** | | | |

Строки таблицы являются записями, а столбцы полями.

**Запись БД** – строка таблицы, содержащая набор значений различных свойств объекта.

**Поле-** -столбец таблиц, включающий все значения определенного свойства.

**Характеристики поля :**

**- имя (название свойства)**

**- тип данных (текстовый, числовой …).**

Как отличить одну запись от другой? В каждой таблице должно быть, по крайней мере, одно ключевое поле, содержимое которого уникально для любой записи в этой таблице. Значения ключевого поля однозначно определяют каждую запись в таблице.

**Первичным ключом** в базе данных называют поле (или совокупность полей), значение которого не повторяется у разных записей.

С каждым полем связано еще одно очень важное **свойство – тип поля. Тип** поля определяет множество значений, которые может принимать данное поле в различных записях.

В реляционных базах данных используются **четыре основных типа поля: числовой; символьный; дата; логический.**

От типа величины зависят те действия, которые можно с ней производить.

**Архитектура Microsoft Access**

Microsoft Access называет объектами все, что может иметь имя (в смысле Access). В базе данных Access основными объектами являются **таблицы, запросы, формы, отчеты, макросы и модули**.

**1. Таблица**. Объект, который определяется и используется для хранения данных. Каждая таблица включает информацию об объекте определенного типа, например о клиентах. Таблица содержит поля (столбцы), в которых хранятся различного рода данные, например фамилия или адрес клиента, и записи (которые называются также строками). В записи собрана вся информация о некотором объекте (человеке, образце продукции и т.п.). Для каждой таблицы можно определить первичный ключ (одно или несколько полей, содержащих уникальные для каждой записи значения) и один или несколько индексов, помогающих ускорить доступ к данным.

**2. Запрос.** Объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать бланк QBE (запрос по образцу) или инструкции SQL (структурированный язык запросов). Можно создать запросы на выборку, обновление, удаление или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц.

**3. Форма.** Объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения. Формы используются для того, чтобы реализовать требования пользователя к представлению данных из запросов или таблиц. Формы можно также распечатать. С помощью формы можно в ответ на некоторое событие, например изменение значения определенных данных, запустить макрос или процедуру VBA.

**4. Отчет**. Объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

**5. Макрос**. Объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в основной форме открывает другую форму. С помощью другого макроса можно осуществлять проверку значения некоторого поля при изменении его содержимого. В макрос можно включить дополнительные условия для выполнения или невыполнения тех или иных указанных в нем действий. Из одного макроса можно также запустить другой макрос или процедуру VBA.

**6. Модуль.** Объект, содержащий программы, написанные на языке Visual Basic для приложений. Модули могут быть независимыми объектами, содержащими функции, вызываемые из любого места приложения, но они могут быть и непосредственно «привязаны» к отдельным формам или отчетам для реакции на те или иные происходящие в них изменения.

***В таблицах хранятся данные, которые вы можете извлекать с помощью запросов. Используя формы, вы можете выводить данные на экран или изменять их. Заметим, что формы и отчеты получают данные как непосредственно из таблиц, так и через запросы.***

С помощью макросов и модулей вы можете изменять ход выполнения приложения; открывать, фильтровать и изменять данные в формах и отчетах; выполнять запросы и создавать новые таблицы.

**Типы данных**

**Числовой формат**

Числовые поля обладают, в основном, теми же свойствами, что и текстовые, за некоторыми отличиями. Тип числового поля может принимать значения из следующего списка:

*Байт – целые числа в пределах от 0 до 255;*

*Целое – целые числа от –32768 до 32767;*

*Длинное целое – целые числа от -2147483648 до 2147483647;*

*С плавающей точкой (4 байт) – числа в пределах от –3.402823ЕЗ8 до 3.402823ЕЗ8;*

*С плавающей точкой (8 байт) – числа в пределах от –1.79769313486232ЕЗ08 до 1.79769313486232ЕЗ08;*

**Код репликации** – глобально уникальный идентификатор (GUID) длиной 16 байт.

Формат числового поля может определяться следующими кодовыми символами:

*. – точка используется в качестве десятичного разделителя;*

*, – запятая применяется как разделитель групп разрядов;*

*0 – вывод цифры или нуля, если разряд незначащий; # – вывод цифры;*

*$ – знак доллара;*

*% – вывод числа в процентном формате;*

*Е или е – вывод числа в экспоненциальной форме.*

**Формат «дата/время»**

Для типа данных Дата/Время существует набор форматов поля, приведенный ниже вместе с примерами:

*Полный формат (устанавливается по умолчанию) 15.04.94 05:30:10 РМ*

*Длинный формат даты Среда 15 Апрель 1994*

*Средний формат даты 15-апр-94*

*Краткий формат даты 15.04.94*

*Длинный формат времени 17:30:10*

*Средний формат времени 05:30 РМ*

*Краткий формат времени 17:30*

**Денежный формат**

*По умолчанию денежный формат представляет собой числа, записанные с двумя знаками после запятой, с разделением групп разрядов и обозначением «р.» в конце*. Пользователь может записать в строке Формат поля шаблон для своего формата. Допустим, необходимо вводить суммы в долларах. Тогда надо установить курсор на слово Денежный и заменить его шаблоном формата, например, таким: # ###$.

Этот формат предполагает, что сумма записана как целое число, с разделением групп из трех разрядов пробелами (чтобы легче было отличать единицы от тысяч, а тысячи от миллионов) и знаком доллара в конце числа, как принято в России, но не в США.

**Формат «счетчик»**

*Поля типа «счетчик» выполняют определенную функцию – автоматическую идентификацию записей таблицы.* Поэтому свойств, которые можно было бы изменить, у такого поля немного. Во-первых, счетчик – это всегда число. Во-вторых, наращивание его значения происходит автоматически. Access предлагает два варианта изменения счетчика: последовательный и случайный. Первый вариант удобен тем, что позволяет нумеровать записи. Второй вариант можно использовать, если надо создать систему кодирования записей в таблице, защищенную от неосторожных действий оператора. Если нумеровать записи последовательно (1, 2, 3, и т. д.), то опечатка при вводе может легко превратить один код в другой, и обнаружить такую ошибку будет трудно. Иначе обстоит дело, если коды выбираются случайным образом (например, 1193517479, 442230790, 1742918387, и т. д.). Поскольку случайные значения счетчика представляют собой многоразрядные числа, то вероятность того, что один код будет введен вместо другого, меньше, чем в первом случае. Кроме того, имея список всех кодов, легко проверить сколь угодно большую таблицу на наличие опечаток. Остается добавить, что использовать этот прием можно, если заменить в дальнейшем тип «счетчик» на другой, потому что изменять данные в поле счетчика оператор не может.

**Логический формат**

*Ячейка в поле логического типа может содержать только одно из двух значений: Да или Нет. Access предлагает еще два варианта названий логического типа: Истина/Ложь и Включено/Выключено.* Вне зависимости от того, какой формат поля вы изберете, в таблице логическое поле будет представлено в виде набора флажков. Когда вы щелкаете флажок мышью, это приводит к тому, что в квадрате появляется изображение галочки, что соответствует логическому значению Да ( ). Сброс флажка означает присвоение ячейке поля логического значения Нет. Использование логических полей может быть самым различным, например, в анкетах или для создания элементов управления.

**Поле объекта OLE**

*Поле объекта OLE (Object Linking and Embedding) – последний выбираемый из списка тип поля. Он имеет всего два свойства: подпись и параметр «обязательное поле». Поля данного типа не хранят информацию как таковую, а содержат ссылки на объекты, которые могут быть включены в БД с использованием OLE-протокола обмена данными. Такими объектами могут быть, например, графические файлы. Даже самая лучшая СУБД не в состоянии обеспечить хранение всех существующих на сегодняшний день видов представления информации. Да это и не нужно. Достаточно осуществить связь с приложением, работающим с тем или иным типом файлов. Для этого и существует протокол OLE.*

Для того чтобы внедрить в ячейку поля OLE-объект, надо установить на нее курсор и, нажав правую кнопку мыши, вызвать контекстное меню. В нем выберите команду **Вставить объект.** Программа раскроет окно диалога для выбора типа объекта. Объект в данном случае отождествляется с приложением, в котором он может быть создан или отредактирован. В том случае, когда в этом окне установлен переключатель Создать новый, открывается соответствующее приложение, работа с которым несколько отличается от обычной. Так, пользователь не имеет возможности открыть существующий файл для того, чтобы сделать его объектом. Например, если выбран документ Word и готовый документ уже имеется на диске, то его придется открыть в редакторе отдельно, скопировать в буфер обмена, а затем перейти в окно документа, связанного с Access, и вставить содержимое буфера. Если же объект создается, что называется, «с нуля», то работа в выбранном приложении происходит обычным образом. Команда выхода из приложения тоже видоизменяется. В данном случае уже нельзя, закрыв приложение, оказаться в Windows, а можно только вернуться в Access.

Второй вариант вставки объекта – выбор переключателя Создать из файла в окне Вставка объекта. В этом случае подразумевается, что файл уже существует, и нужно установить с ним связь. Access откроет окно открытия документа и после выбора нужного файла определит, какое приложение необходимо открыть для работы с ним. Необходимо иметь в виду, что одному типу файлов могут соответствовать несколько приложений. Например, файлы с расширением ВМР используются разными графическими редакторами. Выбранное Access приложение запускается, и в нем открывается выбранный пользователем файл. **Внедрение объекта заканчивается после того, как вы закрываете программу.**

**III этап: Практическая основа урока.**

Учащимся на экране показываю основные моменты работы с таблицами и формами, правилами сохранения БД. Затем учащиеся самостоятельно пробуют выполнить упражнения на отработку данных действий.

**Создание и редактирование структуры таблицы.**

**Создание форм и редактирование в режиме «форма»**

**Практическое задание**

Тема: Отработка навыков открытия, просмотра и редактирования БД

1. **Открыть** БД **«Абитуриент.mdb»**.
2. **Открыть** таблицу **«Абитуриент»** для просмотра.
3. **Добавить** в таблицу две новых записи о следующих абитуриентах:

а) Авдеева Ирина Юрьевна, дата рождения - 17.03.1982, пол - ж, школа - 6, подготовительные курсы – нет, математика – 4, информатика – 5, русский язык – 4.

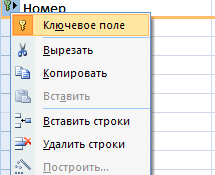
б) Пирогов Анатолий Андреевич, дата рождения - 6.06.1981, пол - м, школа - 63, подготовительные курсы – да, математика – 5, информатика – 4, русский язык – 5.

4. **Заменить** у абитуриентки **Захаровой** оценку по русскому языку на «четвёртку».

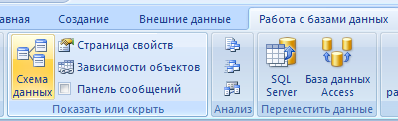
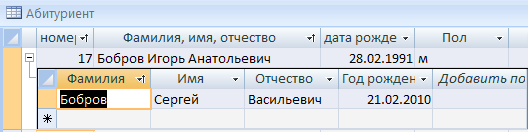
5. **Заменить** у абитуриента **Морозова** номер школы на **31**.

6. **Удалить** записи, содержащие сведения об абитуриентах, не сдававших экзамен по **русскому языку**.

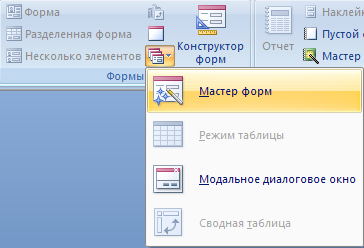
II. Создать еще одну таблицу и дать ей название «Родители». Возьмите поля (Фамилия, Имя, Отчество, год рождения).

Для этого нажмите вкладку «Создать» - выбрать . Далее вводите название полей и **ОБЯЗАТЕЛЬНО прописывайте сразу для каждого поля его свойства.**

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Размер поля |
| Номер | Счетчик | Длинное целое |
| Фамилия | Текстовый | 20 |
| Имя |  |  |
| Отчество |  |  |
| Год рождения | Дата/время | Краткий формат даты |

* Поле «Номер» объявите ключевым (щелкнуть правой кнопкой мыши по названию поля).
* После этого сохраняйте вашу новую таблицу под именем «Родители». Закройте ее.
* Для обеспечения целостности данных выполним заполнение таблицы после установления связи между «Абитуриенты» и «Родители».
* Для этого на вкладке «Работа с базами данных» выбрать «Схема данных»
* Далее «перетащить» в окно схемы данных нужные таблицы ⇒ перетащить одно ключевое поле на другое. В появившемся окне поставить галочку у пункта «обеспечение целостности данных». Появится связь «один к одному». Закройте окно, сохранив изменения.
* После этого зайдите в таблицу «Абитуриенты» и, раскрывая «Плюсики» рядом с фамилией абитуриента, заполните данные о родителях (3-4 записи).

III. Создать форму для имеющихся таблиц с полями: ФИ абитуриента, дата рождения, оценка по математике, ФИО одного из родителей.

* Для этого по аналогии на вкладке «Создать» выбрать «Другие формы» - «Мастер форм».
* Из таблицы «Абитуриент» перенести в выбранные поля: «Фамилия», «Имя», «Отчество», «Подготовительные курсы», а из таблицы «Родители»: «Имя», «Отчество».

(самостоятельно разберитесь с мастером форма. Следуя инструкциям, которые он вам предлагает.)

**Создайте 2 формы разные по стилю и сохраните их под именами «Форма1» и «Форма2»**

**Сохранить полученную БД под именем « новая»**

**Д/З**. Знать основные определения урока.