**(Изучить материал и законспектировать!)**

**Лекция**

**MS Access**

## Понятие базы данных

Хранение информации — одна из важнейших функций компью­тера. Одним из распространенных средств такого хранения являются базы данных. *База данных* — это файл специального формата, содержащий информацию, структурированную заданным образом.

**Структура базы данных**
Большинство баз данных имеют *табличную структуру.* Как мы знаем, в табличной структуре адрес данных определяется пересече­нием строк и столбцов. В базах данных столбцы называются *поля­ми,* а строки — *записями.* Поля образуют *структуру базы данных,* а записи составляют информацию, которая в ней содержится.
Для того чтобы легко усвоить понятие структуры базы данных, надо представить себе пустую базу, в которой пока еще нет никаких данных. Несмотря на то что данных в базе нет, информация в ней все таки есть. Это структура базы, то есть набор полей. Они определяют, что будет записано в эту базу и в каком виде.

**Простейшие базы данных**
Простейшие базы можно создавать, не прибегая к специальным программным средствам. Чтобы файл считался базой данных, информация в нем должна иметь структуру (поля) и быть форма­тирована так, чтобы содержимое соседних полей легко различа­лось. Простейшие базы можно создавать даже в текстовом редакторе Блокнот, то есть обычный текстовый файл при опреде­ленном форматировании тоже может считаться базой данных.
Существует, по крайней мере, два формата текстовых баз данных:

* с заданным разделителем;
* с фиксированной длиной поля.

Несмотря на «примитивность» таких текстовых баз данных, мощные системы управления базами данных позволяют импортировать подобные файлы и преобразовывать их в «настоящие» базы данных. Поэтому если в организации пока нет системы управления базами данных, данные можно хранить в текстовом файле, а потом, когда такая система появится, данные не пропадут и будут успешно импортированы.

 **Свойства полей. Типы полей**
Поля — это основные элементы структуры базы данных. Они обладают *свойствами.* От свойств полей зависит, какие типы дан­ных можно вносить в поле, а какие нет, а также то, что можно делать с данными, содержащимися в поле.
Например, данные, содержащиеся в поле Цена, можно просуммиро­вать, чтобы определить итоговый результат. Суммировать данные, содержащиеся в поле Номер телефона, совершенно бессмысленно, даже если номера телефонов записаны цифрами.

Очевидно, что эти поля обладают разными свойствами и относятся к разным типам.
Основным свойством любого поля является его длина. Длина поля выражается в *символах* или, что то же самое, в *знаках.* От длины поля зависит, сколько информации в нем может поместиться. Мы знаем, что символы *кодируются* одним или двумя байтами, поэтому можно условно считать, что длина поля измеряется в байтах.
Очевидным уникальным свойством любого поля является его *Имя.* Разумеется, одна база данных не может иметь двух полей с одинаковым именем, поскольку компьютер запутается в их содержимом. Но кроме имени у поля есть еще свойство *Подпись.* Подпись — это та информация, которая отображается в заголовке столбца. Ее не надо путать с именем поля, хотя если подпись не задана, то в заголовке отображается имя поля. Разным полям, например, можно задать одинаковые подписи. Это не помешает работе компьютера, поскольку поля при этом по-прежнему сохраняют разные имена. Разные типы полей имеют разное назначение и разные свойства.

1. Основное свойство *текстового поля* — размер.
2. *Числовое поле* служит для ввода числовых данных. Оно тоже имеет размер, но числовые поля бывают разными, например, для ввода *целых чисел* и для ввода *действительных чисел.* В послед­нем случае кроме размера поля задается также размер десятичной части числа.
3. Поля для ввода дат или времени имеют тип *Дата/время.* Для ввода логических данных, имеющих только два значения (Да или Нет; 0 или 1; Истина или Ложь и т. п.), служит специальный тип — *Логическое поле.* Нетрудно догадаться, что длина такого поля всегда равна 1 байту, поскольку этого более чем доста­точно, чтобы выразить логическое значение.
4. Особый тип поля — *Денежный.* Из названия ясно, какие данные в нем хранят. Денежные суммы можно хранить и в числовом иоле, но в денежном формате с ними удобнее работать. В этом случае компьютер изображает числа вместе с денежными едини­цами, различает рубли и копейки, фунты и пенсы, доллары и центы, в общем, обращается с ними элегантнее.
5. В современных базах данных можно хранить не только числа и буквы, но и картинки, музыкальные клипы и видеозаписи. Поле для таких объектов называется *полем объекта OLE.*
6. У текстового поля есть недостаток, связанный с тем, что оно имеет ограниченный размер (не более 256 символов). Если нужно вставить в поле длинный текст, для этого служит поле типа *MEMO.* В нем можно хранить до 65 535 символов. Осо­бенность поля MEMO состоит в том, что реально эти данные хранятся не в поле, а в другом месте, а в поле хранится только указатель на то, где расположен текст.
7. Очень интересно поле *Счетчик.* На первый взгляд это обычное числовое поле, но оно имеет свойство автоматического наращи­вания. Если в базе есть такое поле, то при вводе новой записи в него автоматически вводится число, на единицу большее, чем значение того же поля в предыдущей записи. Это поле удобно для нумерации записей.

## Связанные таблицы

Примеры, которые мы привели выше, можно считать простей­шими базами данных, но на самом деле это не совсем базы, а только таблицы. Если бы информация хранилась в таких простых структурах, то для работы с ней можно было бы обойтись без специальных *систем управления базами данных.* На практике при­ходится иметь дело с более сложными структурами, которые обра­зованы из многих *связанных таблиц.*
Базы данных, имеющие связанные таблицы, называют также ***реля­ционными базами данных***.
Рассмотрим пример работы малого предприятия, занимающегося прокатом компакт-дисков с компьютерными играми. Для того чтобы знать, кто какой диск взял, когда должен возвратить и сколько дисков каждого наименования осталось на складе, пред­приятию необходима база данных. Но если все сведения о поку­пателях и о дисках хранить в одной таблице, то таблица станет очень неудобной для работы. В ней начнутся повторы данных. Всякий раз, когда гражданин Новиков В. П. будет брать очередной диск, придется вписывать его домашний адрес, телефон и паспорт­ные данные. Так никто не работает. Это долго, трудно и чревато многочисленными ошибками.
Гораздо удобнее сделать несколько таблиц. В одной хранить све­дения о клиентах со всеми их паспортными данными, в другой — сведения о выданных дисках, чтобы в любой момент узнать, что выдано клиенту и когда наступает срок возврата, а в третьей табли­це — остаток дисков на складе, чтобы вовремя пополнять запасы. После этого отдельные поля таблиц *связывают.* Если из таблицы Прокат известно, что клиент НВП взял диск D001, то система управления базой данных мгновенно найдет в таблице Клиенты все паспортные данные этого человека, а в таблице Склад все данные об этом диске.
Разделение базы на связанные таблицы не только удобно, но иногда и необходимо. Например, для увеличения числа заказов менеджер фирмы, занимающейся прокатом компакт-дисков, решил поставить в общем зале компьютер, на котором каждый клиент может про­смотреть список имеющихся дисков с иллюстрациями из игр. Если база состоит только из одной таблицы, то вместе с информацией о дисках случайный посетитель получит доступ к информации о других клиентах фирмы. Вряд ли это понравится заказчикам. Такой мене­джер не только не приобретет новых клиентов, но и растеряет тех, которых имел.
Если данные в разных записях начинают повторяться, это может говорить о том, что база имеет плохую структуру. Надо подумать о том, нельзя ли разбить таблицу на группу связанных таблиц
Если заданы связи между таблицами, то работать с разными таблицами можно, как с одной цельной базой данных

**Поля уникальные и ключевые**
Создание базы данных всегда начинается с разработки структуры ее таблиц. Структура должна быть такой, чтобы при работе с базой требовалось вводить в нее как можно меньше данных. Если ввод каких-то данных приходится повторять неоднократно, базу делают из нескольких связанных таблиц. Структуру каждой таблицы раз­рабатывают отдельно.
Для того чтобы связи между таблицами работали надежно, и по записи из одной таблицы можно было однозначно найти записи в другой таблице, надо предусмотреть в таблице *уникальные* поля.
Уникальное поле — это поле, значения в котором не могут повто­ряться.
Если из таблицы Прокат известно, что клиент Новиков просрочил возврат взятого диска, то он должен уплатить штраф. Но в таблице Клиенты фирмы может быть несколько разных Новиковых, и компьютер не разберется, кто же из них должен платить штраф. Это означает, что поле Фамилия не является уникальным и потому его нельзя использовать для связи между таблицами. Поле номера телефона — более удачный кандидат на звание *уни­кального поля,* но, как вы понимаете, и одним телефоном могут пользоваться несколько разных людей.
Если ни одно поле таблицы не приемлемо в качестве уникального, его можно создать искусственно. В нашем примере в таблице Кли­енты фирмы создано поле Шифр, которое образовано первыми тремя буквами фамилии и последними двумя цифрами номера телефона. Его и использовали для связи между таблицами.
Скорее всего, поле Шифр окажется уникальным, и проблем со связями между таблицами не возникнет, но было бы неплохо, если бы компьютер мог просигнализировать в том случае, если вдруг записи в этом поле повторятся. Для этого существует поня­тие *ключевое поле.* При создании структуры таблиц одно поле (или одну комбинацию полей) можно назначить ключевым. С ключевыми полями компьютер работает особо. Он проверяет их уникальность и быстрее выполняет сортировку по таким полям. Ключевое поле — очевидный кандидат для создания связей. Иногда ключевое поле называют *первичным ключом.*
Если при создании таблицы автор не задал ключевое поле, система управления базой данных вежливо напомнит о том, что поле первичного ключа таблице не помешает.
В качестве первичного ключа в таблицах часто используют поле, имеющее тип Счетчик. Ввести два одинаковых значения в такое поле нельзя по определению, поскольку приращение значения поля производится автоматически.
Структура связей между таблицами называется схемой данных.