

MICROSOFT ACCESS. Лекция 2

Таблицы. Создание таблиц

Таблицы — основные объекты базы данных. Без запросов, форм, отчетов и прочего можно обойтись, но если нет таблиц, то данные некуда записывать, а значит, нет и базы. Создание базы начинается с создания первой таблицы.

Создание таблицы состоит в задании ее полей и назначении их свойств. Оно начинается с щелчка на кнопке Создать в окне База данных.

1. Есть несколько способов создания новой таблицы, отличающихся уровнем автоматизации.
2. Самый «автоматичный» способ состоит в импорте таблиц из другой базы, может быть, даже созданной в другой системе. В зависимости от обстоятельств из импортируемой таблицы может поступить структура полей, их названия и свойства, а также и содержимое базы. Если что-то импортируется не совсем так, как надо, необходимые правки (например, в свойства полей) вносят вручную.
3. В тех случаях, когда речь идет о чужой таблице, которая находится на удаленном сервере и которую нельзя импортировать целиком, пользуются режимом Связь с таблицами. Это напоминает подключение к таблице для совместного использования ее данных.
4. Опытные разработчики пользуются Мастером таблиц. Это программа, ускоряющая создание структуры таблицы. Мастер задает ряд вопросов и, руководствуясь полученными ответами, создает структуру таблицы автоматически. Несмотря на то, что этот режим служит для упрощения работы, начинающим пользоваться им не рекомендуется, поскольку, не владея всей терминологией, легко запутаться в вопросах и ответах. Первые таблицы стоит попробовать создать вручную.
5. Пункт Режим таблицы открывает заготовку, в которой все поля имеют формальные имена: Поле1, Поле2... и т. д. и один стандартный текстовый тип. Такую таблицу можно сразу наполнять информацией.
6. Наиболее универсальный ручной метод предоставляет пункт Конструктор. В этом режиме можно самостоятельно задать имена полей, выбрать их тип и настроить свойства.

Для изменения свойств полей надо перейти в режим Конструктор щелчком на кнопке Вид. Чтобы вставить новое поле, надо установить указатель мыши на маркер поля и нажать клавишу INSERT. Чтобы удалить поле, его надо выделить и нажать клавишу DELETE. Закончив создание структуры, можно щелкнуть на кнопке Вид и перейти в Режим таблицы для заполнения ее данными.

Особенности таблиц баз данных

Прежде чем мы приступим к изучению приемов работы с таблицами баз данных, надо обратить внимание на одну особенность всех баз данных, связанную

с сохранением информации. Тех, кто привык работать с другими классами программ, она поначалу обескураживает.

Обычно с документом в программах можно делать все что угодно, пока не настала пора его сохранять. Испортив неаккуратными действиями исходный документ, можно отказаться от сохранения и вернуться к работе с прежней копией. В базах данных это не так.

Таблицы баз данных не являются самостоятельными документами. Сама база — это документ. Ей соответствует файл на диске, и мы можем сделать его копию. Структура таблиц — тоже документ. В некоторых системах она имеет отдельный файл, а в некоторых (например, в Access) такого файла нет, но структура таблиц входит в состав общего файла базы данных наряду с запросами, формами, отчетами и другими объектами. При изменении структуры таблицы система управления базой данных всегда выдает запрос на сохранение изменений.

Но содержание таблиц — это совсем другое дело. Его нельзя сохранить принудительной командой или, наоборот, отказаться от его сохранения. Все изменения в таблицах сохраняются автоматически *в режиме реального времени*. Режим реального времени означает, что, пока мы работаем с таблицей, происходит ее непрерывное сохранение. Как только заканчивается ввод данных в одно поле и происходит переход к следующему полю, данные немедленно записываются на жесткий диск.

Профессионалы высоко ценят эту особенность систем управления базами данных, а начинающих она иногда вводит в заблуждение. Экспериментируя с таблицами, надо знать, что все изменения, которые вносятся в их содержание, имеют необратимый характер. Нельзя что-то изменить, удалить, а потом отказаться от сохранения и вернуться к исходному варианту.

Эта особенность систем управления базами данных требует аккуратного отношения к работе с таблицами. Для экспериментов надо создавать отдельные копии базы или таблиц и работать с ними.

Надежность и безопасность баз данных

Надежность баз данных имеет особую важность. Последствия утраты документа, созданного в текстовом процессоре или графическом редакторе, можно оценить затратами времени, необходимого для его воспроизведения. Утрата базы данных может привести к остановке целой отрасли промышленности и иметь глобальные последствия. Существуют базы данных, от которых зависит движение транспорта, работа банков и промышленных предприятий. Есть базы, содержащие жизненно важные сведения медицинского характера.

Создатели систем управления базами данных не могут полагаться на то, что конкретный пользователь не забудет своевременно дать команду Сохранить. Они учитывают и то, что во время работы может произойти аварийное отключение электричества. Ни при каких условиях информация не должна теряться, поэтому все изменения данных немедленно и автоматически сохраняются на диске.

Совместное использование данных

Системы управления базами данных должны учитывать, что с базами могут одновременно работать много людей. Если бы с базами работали как с документами в текстовом процессоре, то один человек, открывший файл для редактирования, *монополизировал* бы этот файл и *блокировал* бы к нему доступ других пользователей до тех пор, пока файл не будет закрыт и сохранен.

В базах данных один пользователь, вносящий изменения в базу, блокирует только одну запись, с которой он работает, причем ненадолго. Например, известно, что службы автомобильной инспекции имеют базы данных угнанных автомобилей. Тот факт, что где-то в центральной службе идет ввод новых записей об угнанных автомобилях, не мешает инспекторам на местах обращаться к базе по компьютерной сети и наводить необходимые справки. Как только ввод очередной записи завершается, она становится доступной всем инспекторам для просмотра, а некоторым (кому это положено по должности) и для редактирования.

Если в локальной или глобальной сети с одной базой работают несколько пользователей, то каждый может видеть в режиме реального времени те изменения, которые вносят в базу его коллеги.

Приемы работы с таблицами баз данных

С базой данных можно работать обычными приемами управления с помощью мыши.

1. Строка состояния в нижней части окна в Access называется *полем номера записи*. Это поле содержит *кнопки перехода*, с помощью которых можно эффективно перемещаться по таблице.
2. Каждая запись имеет слева кнопку (*маркер записи*). Щелчок на этом маркере выделяет всю запись и готовит ее к копированию, перемещению, удалению.
3. Щелчок правой кнопкой на выделенной записи открывает контекстное меню для операций с записью.
4. Маркер, находящийся в левом верхнем углу таблицы, — это *маркер таблицы*. Щелчок на нем выделяет всю таблицу, а правый щелчок открывает контекстное меню для операций с таблицей в целом.
5. Поля базы данных представлены в таблице столбцами. Каждый столбец имеет заголовок, в котором записано имя поля или то значение, которое задано в свойстве Подпись.
6. Если содержимое поля не полностью уместается в ячейке таблицы, столбец можно расширить. При наведении указателя мыши на границу между столбцами указатель меняет форму. Теперь границу можно перемещать методом перетаскивания, а двойной щелчок, выполненный в этот момент, автоматически устанавливает ширину столбца равной длине самого длинного значения в данном поле.
7. Щелчок на заголовке столбца выделяет весь столбец, а щелчок правой кнопкой на выделенном столбце открывает контекстное меню. В нем есть

очень интересные пункты, позволяющие отсортировать записи по данному полю, вставить новый столбец, скрыть столбец и прочее.

8. Скрытый столбец не исчезает из базы, а только перестает отображаться на экране. Чтобы снова его отобразить, надо навести указатель на границу между столбцами в том месте, где был скрыт столбец, и выполнить двойной щелчок. Скрытый столбец опять станет видимым.

Создание связей между таблицами

Основные преимущества систем управления базами данных реализуются при работе не с отдельными таблицами, а с группами взаимосвязанных таблиц. Для создания связей между таблицами СУБД Access имеет специальное диалоговое окно, которое называется Схема данных.

1. Окно Схема данных открывают щелчком на одноименной кнопке панели инструментов или командой Сервис — Схема данных.
2. Если ранее никаких связей между таблицами базы не было, то при открытии окна Схема данных одновременно открывается окно Добавление таблицы, в котором можно выбрать нужные таблицы для включения в структуру межтабличных связей.
3. Если связи между таблицами уже были заданы, то для введения в схему данных новой таблицы надо щелкнуть правой кнопкой мыши на схеме данных и в контекстном меню выбрать пункт Добавить таблицу.
4. Введя в схему данных все таблицы, которые надо связать, можно приступить к созданию связей между полями таблиц.
5. Связь между полями устанавливают путем перетаскивания имени поля из одной в таблицы в другую на соответствующее ему связанное поле.
6. После перетаскивания открывается диалоговое окно Связи, в котором можно задать свойства образующейся связи.
7. Включение флажка Обеспечение условия целостности данных позволяет защититься от случаев удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.
8. Чтобы условие целостности могло существовать, поле основной таблицы должно обязательно быть ключевым и оба поля должны иметь одинаковый тип.
9. Флажки Каскадное обновление связанных полей и Каскадное удаление связанных записей обеспечивают одновременное обновление или удаление данных во всех подчиненных таблицах при их изменении в главной таблице. Если клиент Соколова выйдет замуж и изменит фамилию на Воронову, то придется внести изменение только в поле Фамилия таблицы Клиенты. В прочих таблицах изменения произойдут автоматически.

Диалоговое окно Схема данных наглядно отображает связи между таблицами.

Чтобы удалить связь, надо щелкнуть на линии связи правой кнопкой мыши и воспользоваться командой Удалить контекстного меню

Запросы

Предположим, что на крупном предприятии есть огромная база данных Кадры, содержащая подробнейшие сведения о каждом сотруднике. Кроме формальной информации база может содержать и конфиденциальную, например, сведения о заработной плате. Вся эта информация хранится в базовых таблицах.

Работать с базой данных Кадры могут разные подразделения предприятия, и всем им нужны разные данные. Не все то, что положено знать службе безопасности предприятия, должно быть доступно главному врачу, и наоборот. Поэтому доступ пользователей к базовым таблицам *закрывают*.

Для доступа к данным есть другое, гораздо более гибкое и удобное средство — *запросы*. Для одной и той же таблицы можно создать множество разных запросов, каждый из которых сможет извлекать из таблицы лишь малую часть информации, но именно ту часть, которая в данный момент необходима. У сотрудника бухгалтерии должен быть запрос, который позволит определить, сколько дней в году по болезни отсутствовал тот или иной работник, но у него не должно быть запроса, позволяющего узнать, чем он болел и где лечился, а у главного врача такой запрос быть должен.

В результате работы запроса из общей исходной базы формируется *результатирующая таблица*, содержащая часть общей информации, соответствующую запросу.

Важным свойством запросов является то, что при создании результирующей таблицы можно не только выбирать информацию из базы, но и обрабатывать ее. При работе запроса данные могут упорядочиваться (сортироваться), фильтроваться (отсеиваться), объединяться, разделяться, изменяться, и при этом никаких изменений в базовых таблицах может не происходить.

Результаты обработки сказываются только на содержании результирующей таблицы, а она имеет временный характер, и иногда ее даже называют моментальным снимком.

И еще одним ценным свойством запросов является их способность выполнять *итоговые вычисления*. Запрос может не только выдать результирующую таблицу, но и найти, например, среднее (наибольшее, наименьшее, суммарное и т. п.) значение по какому-то полю.

Начинающих пользователей баз данных часто удивляет отсутствие в таблицах элементарной возможности вставить новую запись между другими. Записи всегда добавляются только в конец базы. На вопрос, почему так происходит, ответ простой: потому что в упорядочении таблиц нет необходимости. Для этого существуют запросы. Совершенно неважно, под каким номером внесена в таблицу та или иная запись. Если нужно видеть ее в строго определенном месте (например, рядом с другими аналогичными), значит нужно создать запрос, который сгруппирует записи по заданному признаку.

Запросы на выборку

Существует немало различных видов запросов, но самые простые из них и, к тому же, используемые наиболее часто — это *запросы на выборку*.

Цель запроса на выборку состоит в создании *результатирующей таблицы*, в которой отображаются только нужные по условию запроса данные из *базовых таблиц*.

Как и другие объекты Access, запросы можно создавать автоматически с помощью Мастера или вручную. И, как обычно, на этапе обучения лучше не пользоваться Мастером, чтобы почувствовать работу с запросами «кончиками пальцев».

Для создания запросов к базам данных существует специальный *язык запросов*. Он называется *SQL (Structured Query Language — структурированный язык запросов)*. К счастью, те, кто пользуются СУБД Access, могут позволить себе не изучать этот язык. Вместо него в Access есть простое средство, которое называется *бланком запроса по образцу*. С его помощью можно сформировать запрос простыми приемами, перетаскивая элементы запроса между окнами.

Выбор базовых таблиц для запроса

1. Создание запроса к базе начинается с открытия вкладки Запросы диалогового окна База данных и щелчка на кнопке Создать.
2. В открывшемся диалоговом окне Новый запрос задают ручной режим создания запроса выбором пункта Конструктор.
3. Создание запроса в режиме Конструктора начинают с выбора тех таблиц базы, на которых будет основан запрос.
4. Выбор таблиц выполняют в диалоговом окне Добавление таблицы. В нем отображаются все таблицы, имеющиеся в базе.
5. Выбранные таблицы заносят в верхнюю половину бланка запроса по образцу щелчком на кнопке Добавить.
6. В окне Добавление таблицы обратите внимание на наличие трех вкладок: Таблицы, Запросы, Запросы и таблицы. Они говорят о том, что запрос не обязательно основывать только на таблицах. Если ранее уже был создан запрос, то новый запрос можно основывать и на нем. Какие именно таблицы использовать в качестве базовых, решает сам создатель запроса.

Заполнение бланка запроса по образцу

Бланк запроса по образцу — удивительно изящное и удобное средство создания запросов.

1. Бланк запроса по образцу имеет две панели. На верхней панели расположены списки полей тех таблиц, на которых основывается запрос.

2. Строки нижней панели определяют структуру запроса, то есть структуру результирующей таблицы, в которой будут содержаться данные, полученные по результатам запроса.
3. Строку Поле заполняют перетаскиванием названий полей из таблиц в верхней части бланка. Каждому полю будущей результирующей таблицы соответствует один столбец бланка запроса по образцу.
4. Строка Имя таблицы заполняется автоматически при перетаскивании поля.
5. Если щелкнуть на строке Сортировка, появится кнопка раскрывающегося списка, содержащего виды сортировки. Если назначить сортировку по какому-то полю, данные в результирующей таблице будут отсортированы по этому полю.
6. Бывают случаи, когда поле должно присутствовать в бланке запроса по образцу, но не должно отображаться в результирующей таблице. В этом случае можно запретить его вывод на экран, сбросив соответствующий флажок.
7. Самая интересная строка в бланке запроса по образцу называется Условие отбора. Именно здесь и записывают тот критерий, по которому выбирают записи для включения в результирующую таблицу. По каждому полю можно создать свое условие отбора. В нашем примере назначены два условия отбора: повесу игрока (более 80 кг) и по росту (менее 190 см).
8. Запуск запроса выполняют щелчком на кнопке Вид. При запуске образуется результирующая таблица.
9. Чтобы выйти из результирующей таблицы и вернуться к созданию запроса в бланке запроса по образцу, нужно еще раз щелкнуть на кнопке Вид.

Зачем нужен флажок Вывод на экран?

Возникает один закономерный вопрос. Зачем нужен флажок Вывод на экран? Если содержимое поля не надо выводить на экран, то, может быть лучше вообще не включать это поле в бланк запроса по образцу?

Случаи, когда присутствие поля в бланке необходимо, обычно связаны с использованием этого поля для сортировки. Но если при этом сведения в данном поле конфиденциальные, то поле скрывают.

В студии, занимающейся прокатом видеокассет, посетителям могут предложить для просмотра базу видеофильмов, отсортированную в порядке убывания популярности. Чем чаще кассету берут в прокат, тем выше она находится в общем списке.

Но если владелец студии не желает, чтобы любой посетитель мог точно узнать, как часто берутся в прокат те или иные кассеты, то поле, по которому выполнена сортировка, делают скрытым.

Запросы с параметром

Во многих случаях пользователю надо предоставить возможность выбора того, что он хочет найти в таблицах базы данных. Для этого существует специальный вид запроса — *запрос с параметром*.

Предположим, что в базе данных есть таблица, в которой содержатся все результаты чемпионатов мира по футболу. Наша задача — создать запрос, с помощью которого пользователь может определить, в каком году та или иная команда занимала первое место, причем выбор этой команды - его личное дело.

Для этой цели служит специальная команда языка SQL, которая выглядит так: LIKE [...]

В квадратных скобках можно записать любой текст, обращенный к пользователю, например:

LIKE [Введите название страны]

1. Команду LIKE надо поместить в строке Условие отбора и в том поле, по которому производится выбор. В нашем случае это столбец сборных, занимавших первые места в чемпионатах мира по футболу.
2. После запуска запроса открывается диалоговое окно, в котором пользователю предлагается ввести *параметр*. Если в качестве параметра ввести слово Бразилия, то выдается результирующая таблица, содержащая записи по тем чемпионатам, когда сборная Бразилии становилась чемпионом.

Если в качестве параметра ввести слово Италия, то результирующая таблица будет иной.

Разумеется, в нашей небольшой таблице и без запроса нетрудно найти сборные, занимавшие призовые места. Но без запроса не обойтись, если в базе содержатся сотни тысяч записей, причем расположенные в разных таблицах.

Вычисления в запросах

Поле, содержимое которого является результатом расчета по содержимому других полей, называется *вычисляемым полем*.

Вычисляемое поле существует только в результирующей таблице. В исходных (базовых) таблицах такое поле не создается, и при работе обычного запроса таблицы не изменяются. Неправда ли, это очень разумно? Каждый, кто обращается к базе, может с помощью запросов как угодно манипулировать данными и получать любые результаты, но при этом исходные таблицы остаются неизменно одинаковыми для всех пользователей.

1. Для создания запроса, производящего вычисления, служит тот же самый бланк запроса по образцу. Разница только в том, что в одном из столбцов вместо имени поля записывают формулу. В формулу входят заключенные в квадратные скобки названия полей, участвующих в расчете, а также знаки математических операций, например, так:

Стоимость:[Количество]*[Цена]

2. В узкий столбец непросто записать длинную формулу, но если нажать комбинацию клавиш SHIFT F2, то открывается вспомогательное диалоговое окно, которое называется **Область ввода**. В нем можно ввести сколь угодно длинную формулу, а потом щелчком на кнопке ОК перенести ее в бланк запроса по образцу.
3. Если включить отображение вычисляемого поля, результаты расчетов будут выдаваться в результирующей таблице.
4. Ничто не мешает сделать вычисляемое поле полем сортировки, чтобы не только получать новые результаты, но и анализировать их.

Итоговые запросы

Запросы позволяют не только отбирать нужную информацию из таблиц и обрабатывать ее путем создания новых (вычисляемых) полей, но и производить так называемые *итоговые вычисления*.

Примером итогового вычисления может служить сумма всех значений в какой-то *группе записей* или их среднее значение, хотя кроме *суммы* и *среднего значения* существуют и другие итоговые функции.

Поскольку итоговые функции для одной записи не имеют смысла и существуют только для *группы записей*, то предварительно записи надо *сгруппировать* по какому-либо признаку.

1. Рассмотрим работу салона, занимающегося продажей подержанных автомобилей. Результаты работы салона за последнюю неделю содержатся в таблице. В ней можно выделить несколько групп по разному признаку. Записи можно сгруппировать по моделям автомобилей (ВАЗ - отдельно и БМВ - отдельно) или по году выпуска (1989, 1993 и т. д.). Для каждой из групп можно провести итоговое вычисление по полю Цена.
2. Итоговые запросы создают на основе известного нам бланка запроса по образцу, только теперь в нем появляется дополнительная строка — Группировка.
3. Для введения этой строки в бланк надо щелкнуть на кнопке Групповые операции на панели инструментов программы Access .

Далее все происходит очень просто.

4. В тех полях, по которым производится группировка, надо установить (или оставить) функцию Группировка.
5. В тех полях, по которым следует провести итоговое вычисление, надо в строке Группировка раскрыть список и выбрать одну из нескольких итоговых функций.
6. Щелчок на кнопке Вид или Запуск запускает запрос и выдает результирующую таблицу с необходимыми итоговыми данными.

7. В строке Группировка можно указать лишь одну итоговую функцию. А как быть, если надо найти и сумму, и среднее, и максимальное значение, и еще что-то? Решение простое: одно и то же поле можно включить в бланк запроса по образцу несколько раз.

Запросы на изменение

Выше мы говорили о том, что все виды *запросов на выборку* создают временные *результатирующие таблицы*. Базовые таблицы при этом не изменяются. Тем не менее, специально для разработчиков баз данных существует особая группа запросов, которые называются *запросами на изменение*. Они позволяют автоматически создавать новые таблицы или изменять уже имеющиеся. Логика использования запросов на изменение такая:

- создается запрос на выборку, который отбирает данные из разных таблиц или сам создает новые данные путем вычислений;
- после запуска запроса образуется временная результирующая таблица;
- данные из этой временной таблицы используют для создания новых таблиц или изменения существующих.

Существует несколько видов запросов на изменение. Самый простой и понятный — это *запрос на создание таблицы*. Вернемся к примеру с расчетом среднего количества забитых мячей.

1. Предположим, что разработчик таблицы Итоги по командам захотел включить в нее поле Результативность. Конечно, он может рассчитать среднее количество мячей, забитых за игру каждой командой, но если ввести в таблицу такое поле, то его придется заполнять его вручную. Для таблиц, содержащих много записей, это решение неприемлемо.
2. Проще создать запрос на выборку, в который войдут все поля базовой таблицы плюс новое вычисляемое поле.
3. Щелчок на кнопке Вид позволяет убедиться, что запрос работает как положено и создает результирующую таблицу, более полную чем базовая. Теперь можно дать команду на создание новой базовой таблицы, равной результирующей.
4. Эта команда находится в меню Запрос, которое доступно только в режиме Конструктора.
5. В том же меню присутствуют команды для создания запросов на обновление данных, на добавление записей и на удаление записей. Все они относятся к запросам на изменение и работают аналогично, изменяя базовые таблицы в соответствии с данными результирующих таблиц.

Формы

Обычно разработчик базы данных создает структуру таблиц и запросов, но заполнением таблиц информацией он не занимается. Для этого есть специальные

кадры (обычно малоквалифицированные), выполняющие функции наборщиков. Для упрощения их труда разработчик базы может подготовить специальные объекты — *формы*.

Форма представляет собой некий электронный бланк, в котором имеются поля для ввода данных. Наборщик вводит данные в эти поля, и данные автоматически заносятся в таблицы базы.

Зачем нужны формы?

Данные в таблицу можно вносить и без помощи каких-либо форм, но существуют, по крайней мере, четыре причины которые делают формы незаменимым средством ввода данных в базу.

1. Малоквалифицированному персоналу нельзя предоставлять доступ к таблицам (самому ценному из того, что есть в базе) Представьте, что будет, если новичок «наведет порядок» в таблице банка, хранящей расчетные счета клиентов.
2. Разные люди могут иметь разные права доступа к информации, хранящейся в таблицах. Например, один имеет право вводить только имена и адреса клиентов, другой – только номера их расчетных счетов, а третий — только денежные суммы, хранящиеся на этих счетах. Сговор между этими людьми должен быть исключен. Для ввода данных им предоставляют разные формы, хотя данные из форм могут поступать в одну таблицу.
3. Ввод данных в таблицу – чрезвычайно утомительное занятие. Уже после нескольких часов работы люди делают ошибки. Ввод данных в форму проще. Здесь многое можно автоматизировать. К тому же, элементы управления форм настраивают таким образом, чтобы при вводе данных выполнялась их первичная проверка.
4. Информацию для баз данных берут из бумажных бланков: анкет, заявлений, накладных, счетов, описей, ведомостей, справок. Экранные формы можно сделать точной копией бумажных бланков, с которых происходит ввод данных. Благодаря этому во много раз уменьшается количество ошибок при вводе и значительно снижается утомляемость персонала.

Создание форм

Как и другие объекты Access, формы можно создавать вручную или автоматически, причем несколькими способами. При создании таблиц и запросов мы рекомендовали на первых порах автоматическими средствами не пользоваться, чтобы вникнуть в терминологию и подготовить себя к работе с Мастером, задающим непонятные для начинающих вопросы. С формами дело обстоит иначе. Они состоят из многочисленных элементов управления, и от того, насколько аккуратно эти элементы расположены на экране, зависит внешний вид формы.

Автоматические средства позволяют создавать аккуратные формы и не задают пользователю лишних вопросов. Начинать работу лучше с них.

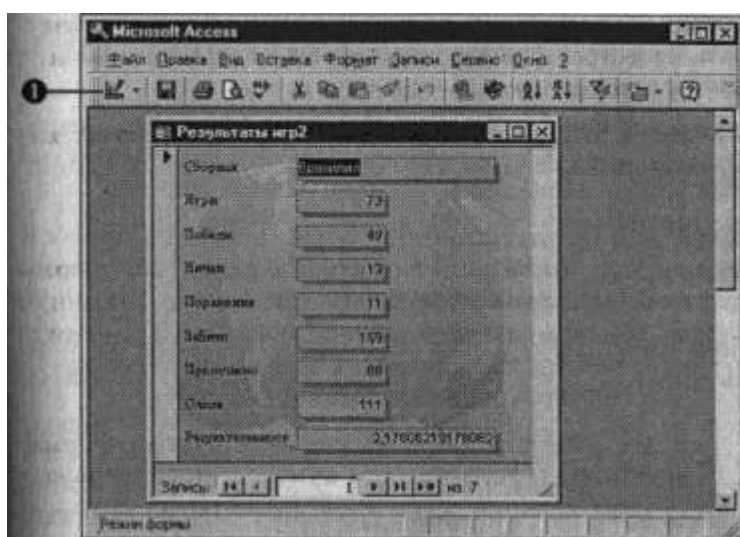
Автоформы

Автоформы — самый простой вид автоматических форм. Для создания автоформы надо открыть вкладку Формы в диалоговом окне База данных и щелкнуть на кнопке Создать — откроется окно Новая форма.

В диалоговом окне Новая форма выбирают в качестве источника данных для формы какую-либо таблицу или запрос, после чего создают автоформу двойным щелчком в списке выбора вида автоформы (табличная, ленточная или в столбец).

Создание формы с помощью Мастера

1. С помощью Мастера форма создается всего в четыре этапа:
 1. выбор полей, данные для которых можно будет вводить в форме;
 2. выбор внешнего вида формы (один из четырех);
 3. выбор фоновой рисунка формы (один из десяти);
 4. задание имени формы.
5. Все эти пункты достаточно хорошо объяснены в Мастере и не требуют никаких пояснений.
2. Готовую форму можно сразу же использовать для просмотра существующих записей или для ввода новых.



Панель

элементов



Структура форм

Создавая формы автоматическими средствами, можно не задумываться над их структурой, но при разработке формы вручную со структурой приходится иметь дело.

Структуру формы составляют ее *разделы*, а разделы содержат *элементы управления*.

Разделы формы

1. Самый простой способ познакомиться с разделами формы состоит в том, чтобы взять готовую форму, например, созданную с помощью Мастера, и посмотреть ее устройство в режиме Конструктора. Как мы уже знаем, для этого надо щелкнуть на кнопке Вид на панели управления Access .
2. При просмотре в Конструкторе мы видим структуру формы. Обратите внимание на то, что рядом с ней открывается *панель элементов*, содержащая заготовки и инструменты для создания элементов управления формы.
3. В структуре формы четко видны три раздела: *раздел заголовка* формы, *область данных* и *раздел примечания* формы.

В нашем случае заполнена только область данных. Так произошло потому, что форму создавал Мастер, который не потрудился создать и заполнить вспомогательные разделы.

4. Все, что содержится в области данных, является *элементами управления*. В нашем случае здесь присутствуют элементы управления только двух типов: *связанное поле* (то, что в него вводится, поступает и в одноименное поле таблицы, на базе которой создана форма) и *присоединенная надпись* (называется так, поскольку перемещается вместе со своим элементом управления). В нашем случае содержание присоединенной

надписи совпадает с названием связанного поля, но, как вы понимаете, это можно и изменить.

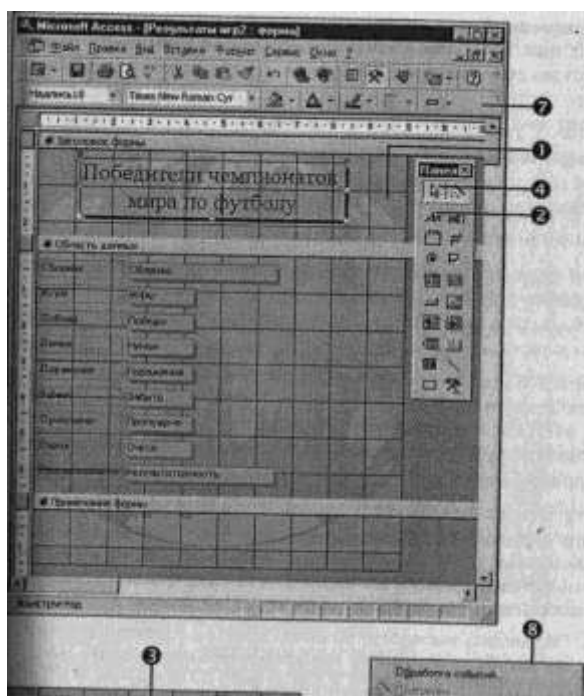
5. Фоновый рисунок, лежащий под элементами управления, показывает размер *рабочего поля формы*.
6. Размеры разделов и размеры рабочего поля формы можно изменять с помощью мыши. При наведении на границу раздела указатель меняет форму. В этот момент границу можно перемещать методом перетаскивания.

Создание надписей

Редактирование форм состоит в создании новых или изменении имеющихся элементов управления, а также в изменении их взаимного расположения.

При рассмотрении приемов создания новых элементов управления мы воспользуемся тем фактом, что Мастер, создавший форму, не заполнил ее раздел заголовка.

1. Перетащив вниз разделительную границу между заголовком и областью данных, мы можем освободить вверху достаточно места для создания крупной надписи.

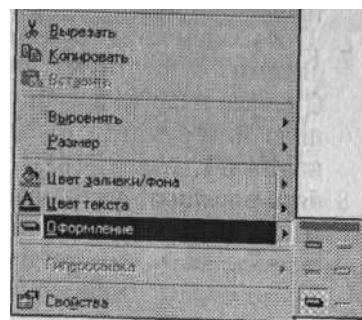


2. На панели элементов существует специальный элемент управления для создания заголовков, который называется Надпись.

3. Щелкнув на нем, а потом на форме, мы получаем текстовую рамку, в которую можно вводить произвольный текст. При вводе текста не надо заботиться о его форматировании. Неважно, как он выглядит и где расположен.

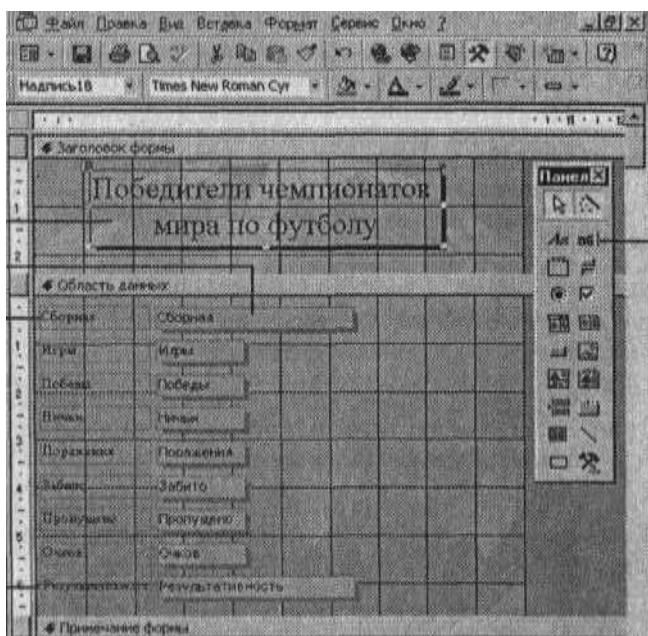
Закончив ввод, надо нажать клавишу ENTER,

4. Для форматирования элемента управления его надо сначала выделить. Для этого служит инструмент Выбор объектов.
5. При выделении элемента управления вокруг него образуется рамка с восемью маркерами (по углам и по центрам сторон рамки). Рамку можно растягивать или сжимать методом перетаскивания



- границ. При наведении на маркер указатель мыши меняет форму, принимая изображение открытой ладони. В этот момент рамку можно перемещать.
- Особую роль играет левый верхний маркер рамки. При наведении на него указатель мыши принимает форму указательного пальца. О роли этого маркера мы расскажем чуть позже.
 - Когда объект выделен, можно изменять параметры шрифта, метод выравнивания текста и другие элементы форматирования. Это выполняют обычными средствами форматирования, доступными через соответствующую панель инструментов Access.
 - Если щелкнуть на выделенном элементе правой кнопкой мыши, откроется его контекстное меню, в котором имеются дополнительные возможности изменения оформления. В нашем случае, например, применено Оформление с тенью

Создание и редактирование связанных полей



1 Заголовок таблицы, который мы только что создали, не связан ни с одним из полей таблицы. Поэтому элемент управления Надпись еще называют *свободным полем*. Текст, введенный в него, остается неизменным независимо от того, какую запись в этот момент просматривают в форме.

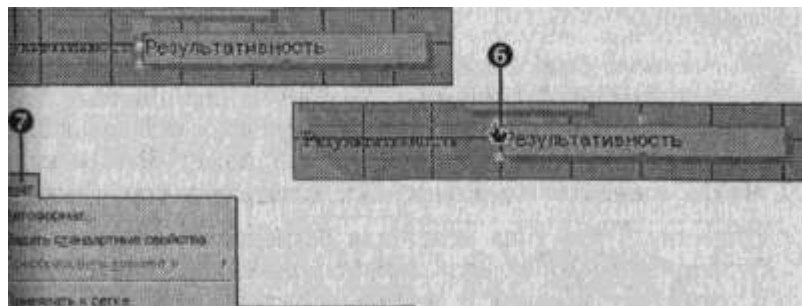
2. Совсем иначе обстоит дело с элементами управления, в которых отображается содержимое полей таблицы. Такие элементы управления называют *связанными полями*.

3. Для их создания служит элемент

Поле на панели элементов.

- При создании связанного поля вместе с ним одновременно образуется еще один элемент управления – присоединенная надпись. Она перемещается вместе со связанным полем и образует с ним единое целое.
- Обратите внимание на то, что, что слово «Результативность» в присоединенной надписи записано без последней буквы. Это не ошибка. Просто Мастер, создававший форму, сделал это неаккуратно, и связанное поле «наехало» на присоединенную надпись.
- Оторвать поле от присоединенной надписи позволяет уже упомянутый маркер, расположенный в левом верхнем углу. При наведении на него указатель мыши принимает форму указательного пальца. В этот момент связанное поле можно оторвать от присоединенной надписи и перемещать отдельно.

5. Перемещать элементы управления и изменять их размеры с помощью мыши не слишком удобно. Гораздо удобнее использовать для этой цели курсорные клавиши в комбинации с клавишами SHIFT или CTRL. В первом случае происходит изменение размеров элемента управления, а во втором – изменение его расположения.



6. Чтобы элементы управления располагались в форме ровными рядами, существуют специальные команды выравнивания. Сначала надо выделить группу элементов управления с помощью инструмента Выбор объектов (группа выбирается при нажатой клавише SHIFT), а потом дать команду Формат ► Выровнять и выбрать метод выравнивания.

Прочие элементы управления формы

При создании формы вручную элементы управления размещают на ней так, как удобно проектировщику. Созданные элементы управления формы выравнивают с помощью команды Формат ► Выровнять.

Кроме рассмотренных выше элементов управления Надпись и Поле, существует еще несколько полезных элементов управления.

1. *Переключатели (радиокнопки).* С ними можно связать команды, например, выполняющие фильтрацию.

2. *Флажки.* Действуют аналогично переключателям, но в отличие от них, допускают множественный выбор. Удобны для управления режимами сортировки данных.

3. *Список.* Может содержать фиксированный набор значений или значения из заданного поля одной из таблиц. Позволяет не вводить данные, а выбирать их из списка.

4. *Поле со списком.* Применяется так же, как и список, но занимает меньше места в форме, поскольку список открывается только после щелчка на раскрывающей кнопке.

5. *Командные кнопки.* С каждой из них можно связать какую-либо полезную команду, например команду поиска записи, перехода между записями и другие.

6. *Вкладки.* Позволяют разместить много информации на ограниченной площади. На вкладках размещают другие элементы управления.
7. *Поле объекта OLE.* Служит для размещения внешнего объекта, соответствующего принятой в Windows концепции связывания и внедрения объектов. Объектом, как правило, является иллюстрация, например фотография, но это может быть и видеозапись, и музыкальный фрагмент, и голосовое сообщение.

Существуют два типа полей для размещения объектов OLE: Свободная рамка объекта и Присоединенная рамка объекта. В первом случае, рамка не связана ни с каким полем таблиц базы данных. Объект, находящийся в ней, выполняет роль иллюстрации и служит для оформления формы. С присоединенной рамкой связано одно из полей таблицы. В ней отображается содержимое этого поля. Это содержимое может меняться при переходе от одной записи к другой.

Отчеты

Напомним функции основных объектов *базы данных*:

- таблицы служат для хранения данных;
- запросы служат для выбора данных из таблиц, а также для автоматизации операций по обновлению и изменению таблиц;
- формы служат для упрощения операций ввода данных в таблицы, но могут быть использованы и для просмотра результатов работы запросов на экране.

Из основных объектов нам осталось рассмотреть только *отчеты*. Отчеты во многом похожи на формы и тоже позволяют получить результаты работы запросов в наглядной форме, но только не на экране, а в виде распечатки на принтере. Таким образом, в результате работы отчета создается *бумажный документ*.

Автоотчеты

Большая часть того, что было сказано о формах, относится и к отчетам. Выбрав в диалоговом окне База данных вкладку Отчеты и щелкнув на кнопке Создать, мы получаем диалоговое окно Новый отчет, позволяющее создать отчет автоматически (*автоотчет*), с помощью Мастера или вручную. Точно так же, как и с формами, с отчетами удобнее знакомиться в режиме автоматического создания. Создайте на основе любой таблицы автоотчет в *столбец* или *ленточный*. Операция настолько проста, что сводится к одному щелчку левой кнопки мыши.

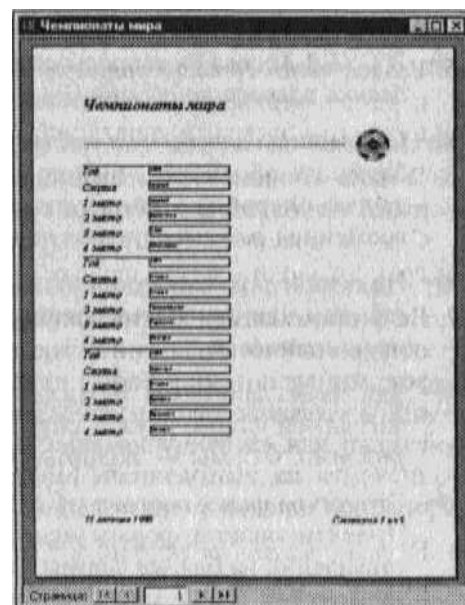
Отчеты предназначены для вывода информации на принтер, поэтому для расчета расположения данных на печатной странице программа Access должна

«знать» все необходимое об особенностях принтера. Эти данные Access получает от операционной системы. Соответственно, принтер в системе должен быть установлен. При отсутствии принтера отчеты создавать все-таки можно. Достаточно выполнить программную установку с помощью команды операционной системы: Пуск ► Настройка Принтеры ► Установка принтера, после чего зарегистрировать драйвер принтера, либо взяв его с гибкого диска, либо выбрав один из драйверов, прилагающихся к самой операционной системе.

Структура отчета

Как и формы, отчеты состоят из разделов, а разделы могут содержать элементы управления. Но, в отличие от форм, разделов в отчетах больше, а элементов управления, наоборот, меньше. Со структурой отчета проще всего ознакомиться, создав какой-либо автоотчет, а затем открыв его в режиме Конструктора.

1. Структура отчета состоит из пяти разделов: *заголовка отчета*, *верхнего колонтитула*, *области данных*, *нижнего колонтитула* и *примечания отчета*. По сравнению с формами новыми являются разделы верхнего и нижнего колонтитулов.
2. Раздел заголовка служит для печати общего заголовка отчета.
3. Раздел верхнего колонтитула можно использовать для печати подзаголовков, если отчет имеет сложную структуру и занимает много страниц. Здесь можно также помещать и колонцифры (номера страниц), если это не сделано в нижнем колонтитуле.
4. В области данных размещают элементы управления, связанные с содержимым полей таблиц базы. В эти элементы Управления выдаются данные из таблиц для печати на принтере. Порядок размещения и выравнивания элементов управления тот же, что и при создании структуры форм.
5. Раздел нижнего колонтитула используют для тех же целей что и раздел верхнего колонтитула. В нашем случае в нем размещены два элемента управления.
6. В первом элементе управления выводится текущая дата. Для этого использована встроенная в Access функция Now(). Она возвращает текущую дату и помещает ее в поле, а отчет воспроизводит ее при печати.
7. Во втором элементе управления выводится номер страницы и общее количество страниц. Для их определения использованы встроенные функции Page() и Pages(). Тот текст, который записан в кавычках, воспроизводится «буквально», а оператор & служит для «склеивания»



текста, заключенного в кавычки, со значениями, возвращаемыми функциями. Оператор & называется оператором конкатенации.

8. Раздел примечания используют для размещения дополнительной информации. В нашем примере он не использован.

Закрепление пройденного

Системы управления базами данных (СУБД) — это специальные программные средства, предназначенные для работы с файлами баз данных (файлами специального формата, содержащими информацию, структурированную заданным образом).

Современные СУБД позволяют хранить в виде файлов данные любых типов: числовые, текстовые, графические, звуковые, видео и прочие.

Данные в базах хранятся в виде *таблиц*. Каждая таблица имеет *структуру*.

Структура таблицы определяется составом ее *полей* и их *свойствами*. Важнейшими свойствами полей являются: *тип поля* и *размер поля*. Для хранения разных *типов данных* используют *поля соответствующих типов*.

Данные, хранящиеся в таблице, можно изменять, удалять, сортировать, фильтровать, размножать и выполнять с ними другие операции.

Для автоматизации операций по работе с данными, в частности, для отбора нужных данных, применяют специальные объекты, которые называются *запросами*.

В СУБД Access запросы создают с помощью специального *бланка запроса по образцу*.

С помощью *запросов на выборку* производят выбор данных из базы, их обработку, выполнение итоговых вычислений и другие операции. По результатам работы запроса создается временная *результатирующая таблица*.

На основе результирующей таблицы, можно создавать новые таблицы или изменять существующие. Для этого служат *запросы на изменение*.

Для ввода данных в таблицы или для просмотра данных в наглядной форме служат специальные объекты, называемые *формами*. Формы — экранные объекты.

Структура форм состоит из *разделов* и *элементов управления*. Проектирование формы состоит в размещении элементов управления на бланке формы и в задании связей между этими элементами и полями таблиц или запросов базы данных.

Создание форм можно выполнять автоматически (*автоформы*), полуавтоматически (с помощью *Мастера*) или вручную (в режиме *Конструктора*).

Размещение элементов управления на бланке формы автоматизировано. В большинстве случаев при создании нового элемента запускается *программа-Мастер*, с помощью которой происходит настройка свойств элемента управления.

Для создания печатных документов, которые содержат информацию из базовых таблиц или из результирующих таблиц, полученных по результатам работы запросов, служат специальные объекты — *отчеты*.

Отчеты отличаются от форм тем, что предназначены не для ввода данных, а только для вывода, а также тем, что создают не экранные, а печатные документы.

Структура отчетов, как и форм, состоит из разделов и элементов управления. Проектирование отчета состоит в создании структуры его разделов и в размещении элементов управления внутри этих разделов, а также в задании связей между этими элементами и полями таблиц или запросов базы данных.

Создание отчетов может выполняться автоматически (*автоотчеты*), полуавтоматически (с помощью *Мастера*) или вручную (в режиме *Конструктора*).

Таблицы, запросы, формы и отчеты являются основными объектами базы данных. Их разрабатывает разработчик базы. Пользователь базы использует эти объекты без вмешательства в их структуру.

Разработчик базы данных имеет также два типа дополнительных объектов: *макросы* и *модули*. Эти объекты создают в тех случаях, когда стандартных средств управления базой данных оказывается недостаточно для выполнения операций, необходимых заказчику системы. С помощью *макросов* создают *макрокоманды*, упрощающие наиболее утомительные операции с базой, а с помощью *модулей*, написанных на языке программирования Visual Basic, создают программные процедуры для выполнения нестандартных операций.

Контрольные вопросы

1. В каких *объектах* хранятся данные базы?
2. Чем отличаются *поля* и *записи* таблицы?
3. Какие *типы полей* таблиц вы знаете?
4. Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?
5. Содержит ли какую-либо информацию таблица, в которой нет полей? Может ли существовать такая таблица?
6. В чем состоит особенность поля типа Счетчик?
7. В чем состоит особенность поля типа Мемо?
8. Какое поле можно считать уникальным?
9. Каким должно быть поле первичного ключа?
10. В чем преимущество поля, являющегося первичным ключом?
11. Какую базу данных называют реляционной?
12. Поле какого типа является естественным кандидатом на роль ключевого поля?
13. В каком диалоговом окне создают связи между полями таблиц базы данных?
14. Почему при закрытии таблицы программа Access не предлагает выполнить сохранение внесенных данных?
15. Для чего предназначены запросы?
16. Запросы баз данных обращены к таблицам. Как называются таблицы, на базе которых основан запрос?

17. В результате работы запроса образуется таблица. Как называется эта таблица?
18. Как называется категория запросов, предназначенная для выбора данных из таблиц базы?
19. Как называется бланк, заполнением которого создается структура запроса?
20. Как называется поле, данные в котором образуются в результате расчета с участием данных, содержащихся в других полях?
21. Как называется категория запросов, выполняющих итоговые вычисления?
22. Какие итоговые функции вы знаете?
23. Для выполнения итоговых вычислений записи в таблицах должны быть сгруппированы по какому-либо признаку. В то же время, в бланке запроса по образцу нет строки для команды группировки. Как внести в него эту строку?
24. Каждому полю таблицы в бланке запроса по образцу соответствует один столбец. В нем можно дать команду на расчет одной итоговой функции. Как сделать четыре итоговых вычисления по одному полю, например определить суммарную, среднюю, максимальную и минимальную цену товаров в некоей группе? (Задача часто встречается на практике.)
25. В запросе имеется вычисляемое поле с названием ФИО. В нем записано выражение, согласно которому берутся данные из трех полей: Фамилия, Имя и Отчество, потом эти данные сливаются вместе в одну строку и заносятся в это поле. Как сделать, чтобы в результирующей таблице отображалось содержимое поля ФИО без дублирования данных из тех полей, по которым оно рассчитано? (Задача часто встречается на практике.)
26. Как с помощью запроса провести сортировку записей в таблице по какому-либо полю?
27. Какие два типа сортировки вы знаете?
28. В базе данных есть поля: Фамилия, Имя, Отчество. В столбцах бланка запроса по образцу, соответствующих этим полям, включено указание на проведение сортировки. Так по какому же из трех полей на самом деле произойдет сортировка? (В тексте книги об этом не говорилось, но попробуйте догадаться логически и проверить гипотезу на практике).
29. Для чего предназначены формы?
30. Какие методы автоматического создания форм вы знаете?
31. Из чего состоят формы?
32. Какие разделы форм вы знаете?
33. Какие элементы управления форм вы знаете?
34. При создании элемента управления формы рядом с ним автоматически образуется элемент управления, называемый присоединенной надписью. Как оторвать эту надпись от элемента управления, если разработчик не хочет, чтобы она располагалась слева, а желает поместить ее ниже элемента управления?
35. Какой командой можно выровнять элементы управления формы, расположенные неаккуратно?

36. При заполнении таблиц удобно не вводить часто повторяющиеся данные, а выбирать их из заранее заготовленного списка. Это не только облегчает работу наборщиков, но и позволяет избежать ошибок в наиболее ответственных местах. Как называются элементы управления формы, позволяющие это сделать?
37. Можно ли использовать формы не только для ввода, но и для вывода данных? Если да, то на какое устройство компьютерной системы выполняется этот вывод?
38. Для чего предназначены отчеты!
39. На какое устройство компьютерной системы происходит вывод данных при работе отчета?
40. Из чего состоят отчеты?
41. Какие разделы отчетов вы знаете?
42. Что общего и в чем различие между разделами отчетов и разделами форм?
43. Какие способы автоматической генерации отчетов вы знаете?
44. Что следует указать программе Access для автоматической генерации отчета? (На базе чего выполняется отчет?)
45. С какими еще объектами кроме таблиц, запросов, форм и отчетов работает СУБД Access? Для чего они предназначены?