

Основы баз данных

В этой главе...

- Терминология баз данных
- Типы полей
- Плоские и реляционные базы данных
- Построение базы данных
- Добавление и удаление таблиц

Это самая важная глава книги. Прочитав о том, почему Access наиболее подходящий инструмент для работы с базами данных (глава 1), и о том, как работать с графическим интерфейсом Access (глава 2), вы можете приступить к построению базы данных.

Терминология баз данных

Не пугайтесь ужасного слова “терминология”. Вам придется запомнить всего несколько терминов. Этого достаточно, чтобы понимать, о чем идет речь в диалоговых окнах и вкладках Access.

К сожалению, знать термины обязательно. Без них вы слепы и беспомощны. Некоторые термины уже упоминались в предыдущих главах. Мы повторим их в данной главе, чтобы вы лучше усвоили их. Никаких мудреных вещей мы вам не предлагаем. Это всего лишь несколько простых слов, которые вы без труда поймете.

Термины, рассматриваемые в этом разделе, размещены в порядке их “величины”. Начнем с самого маленького понятия — “данные”. Переходя от него к более крупным, закончим самым большим термином — “база данных”. Это сделано специально для того, чтобы большая картина постепенно складывалась у вас в голове из маленьких кусочков. Цель главы — помочь вам увидеть, как эти маленькие кусочки совмещаются друг с другом.

Данные

Данные — это информация, которая хранится в компьютере с помощью программы Access. Данные могут храниться у вас в голове, однако в Access они хранятся иначе. Например, вы можете думать о некотором человеке, что его зовут “Джон Смит”. Возможно, вы не помните его фамилию или в вашем кругу она не упоминается. Тогда вы можете думать, что его зовут просто “Джон”. В вашей голове эти две порции данных эквивалентны и взаимозаменяемы: термины “Джон Смит” и “Джон” обозначают одного и того же человека. В базе данных такая двусмысленность недопустима. Если столбец таблицы называется Имя, в нем должна храниться строка Джон, а не Смит или Джон

СМИТ. Если в таблице есть столбец Имя и фамилия, в нем должна храниться строка Джон Смит. Рекомендуется детализировать данные как можно подробнее. Например, имена и фамилии лучше хранить в двух отдельных столбцах. Тогда будет легче сортировать фамилии по алфавиту или искать человека с определенным именем.

Порции данных логически должны быть как можно более простыми. Вас не должно смущать, что вместо одного столбца появилось два. При необходимости с помощью инструментов Access вы легко объедините имя и фамилию (например, если в отчете их нужно вывести вместе).

Столбец

Данные не должны быть разбросаны как попало. Каждая порция данных должна находиться на своем месте. Для этого компьютерные гуру изобрели понятие *столбца*. Выше уже упоминалось, что таблица состоит из строк и столбцов. Каждый столбец представляет некоторый параметр, характеризующий объект. Например, если в таблице хранятся данные о компаниях, то в первом столбце могут находиться названия компаний, во втором — адреса, в третьем — фамилии генеральных директоров и т.д.

Пересечение столбца и строки называется *полем*. Нельзя не упомянуть о разногласиях и разночтениях, царящих в этой области. В программах управления базами данных FoxPro и FileMaker столбцы называются полями. В более солидных программах, таких как Oracle и Microsoft SQL Server, столбцы называются столбцами. Программа Access занимает промежуточное положение: в некоторых местах интерфейса упоминаются столбцы, а в некоторых — поля. Многие люди считают, что это не имеет никакого значения, и нудные педанты вроде нас напрасно придираются к словам. В самом деле, когда вы увидите в контекстном меню Access команду Создать поле, то разве не понятно, что имеется в виду добавление столбца в таблицу? Конечно же, понятно, но только для опытного пользователя, который, поработав с базами данных, привык в этом и перестал замечать несоответствие между словом и делом. Наш опыт преподавания свидетельствует о том, что начинающих это часто сбивает с толку. Итак, запомните: пересечение столбца и строки называется “полем”, однако “полем” часто называют также столбец.

Запись

Поле — прекрасное место для хранения порции данных. Однако поля нужно каким-либо образом сопоставлять друг с другом. Откуда нам известно, что имя и фамилия принадлежат одному и тому же человеку? Что-то должно объединять поля. Их объединяет *запись*. Если в таблице хранятся сведения о двух компаниях, то хранятся они в двух записях, причем в каждой записи находится информация об одной компании. Информация о 50 компаниях хранится в 50 записях.

Размышляя о значении термина “запись”, учитывайте следующие факторы.

- ✓ Запись часто называют *строкой*. В программах FoxPro и FileMaker используется термин “запись”, а в Oracle и Microsoft SQL Server — “строка”. Программа Access опять же находится в промежуточном положении: в некоторых местах интерфейса написано “запись”, а в некоторых — “строка”. К счастью, в этом случае неразберихи нет. Термины “запись” и “строка” — полные синонимы. Однако не путайте строки таблицы с текстовыми строками, которые могут быть расположены во многих местах: в полях таблицы, полях формы, областях отчета и т.д.

- ✓ Каждая запись таблицы состоит из того же набора полей, что и любая другая запись этой же таблицы. Естественно, данные в полях разные (хотя, это не обязательно; например, несколько человек могут иметь одно и то же имя). Некоторые поля могут быть пустыми; например, если у человека нет телефона, то соответствующее поле в его записи остается пустым.
- ✓ Одна запись содержит всю информацию о конкретном объекте. Естественно, всю доступную и уместную информацию.

Таблица

Таблица — это набор записей. Каждая запись содержит информацию о конкретном объекте. Обратите внимание: расположение данных в каждой записи определяется последовательностью столбцов таблицы. Записи соответствуют единственному числу, а таблице — множественное. Например, информация о **компаниях** хранится в записи, а о **компаниях** — в таблице. В каждой таблице хранится информация только об одном классе объектов, например, в таблице, посвященной компаниям, не могут храниться данные о заказах, даже если они сделаны этими компаниями; информация о заказах должна храниться в отдельной таблице.

Предположим, у нас есть таблица, в которой хранится информация о компаниях, и мы решили добавить в нее столбец, в который будут записываться заказы, сделанные этой компанией. Казалось бы, база данных будет проще: вместо двух таблиц у нас будет одна. Однако, если нам понадобится составить отчет, как мы будем извлекать информацию о заказах? Одна компания может сделать несколько заказов. Если все заказы компании расположены в одном поле, Access не сможет работать с ними. Таблица заказов содержит много столбцов: дата заказа, цена, способ доставки, выполнен ли заказ и т.д. Представьте себе, что все это нужно разместить в одном поле, причем не для одного заказа, а для многих (может быть, даже для сотен заказов). Это будет больше похоже на длинный рассказ в произвольной форме, а не на четкую информацию. Следовательно, такая совмещенная таблица вряд ли может быть полезной.

База данных

База данных — это набор таблиц. Информация хранится в таблицах. Однако кроме таблиц база данных содержит много других компонентов, облегчающих работу с таблицами. Поэтому короткое определение, приведенное в начале абзаца, нуждается в дополнении: база данных содержит таблицы, запросы, отчеты и формы. Где находятся все эти компоненты? Не могут же они быть разбросаны по всему жесткому диску. Действительно, все компоненты базы данных хранятся в одном файле. Часто встречается термин *файл базы данных*. В большинстве случаев это то же самое, что и база данных. Одна база данных хранится в одном файле. Имя базы данных совпадает с именем файла.

Следовательно, база данных — это больше, чем просто данные. Это еще и средства работы с данными (инструменты ввода, сортировки, управления, форматирования, вывода), без которых совладать с огромным объемом информации было бы просто невозможно.

Типы полей

В поле, как вы уже знаете, хранится порция данных. В каждом поле находится единственная элементарная порция информации, например, имя человека, дата заказа или цвет обоев.

В нашем бесконечно многообразном мире существует огромное количество типов информации. Для их хранения в базах данных Access предусмотрено десять типов столбцов (или, что то же самое, полей):

- ✓ Текстовый (Text);
- ✓ Поле МЕМО (Memo);
- ✓ Числовой (Number);
- ✓ Дата/Время (Date/Time);
- ✓ Денежный (Currency);
- ✓ Счетчик (AutoNumber);
- ✓ Логический (Yes/No);
- ✓ Объект OLE (OLE Object);
- ✓ Гиперссылка (Hyperlink);
- ✓ Вложение (Attachment).



Пока что не пытайтесь догадаться о назначении типов, судя по их имени. Немного далее мы подробно расскажем о каждом типе. Как видите, список охватывает почти все типы информации, которые только можно вообразить. Естественно, в нем нет типов информации, которые тяжело формализовать и невозможно хранить в базах данных (воспоминания, тактильные ощущения, запахи, эпитеты и метафоры, философские категории). Однако как же быть с типами, о которых вы точно знаете, что их можно хранить, но которых вы не видите в списке (например, кинофильмы, музыкальные произведения)? Не беспокойтесь, информацию этих типов можно хранить в полях Объект OLE. Не забывайте, что почти каждый тип можно настраивать, чтобы он как можно лучше подходил для решаемой задачи. Например, для типа Числовой можно задать количество разрядов. Более подробно настройка типов рассматривается в главе 4. Обратите также внимание на то, что всегда используется одно и то же имя типа, например, слово Числовой нельзя заменить на числовой, Числовое, величина или число.

Ниже приведено описание десяти типов полей.



- ✓ **Текстовый**. В поле этого типа можно хранить до 255 символов — букв, цифр, знаков препинания и т.д.

Числа в текстовых полях не являются числами. Это всего лишь последовательность цифр, расположенных рядом. Будьте осторожны и не путайте числа с текстом, даже если внешне текст выглядит как число.

- ✓ **Каждое текстовое поле имеет параметр, о котором нельзя забывать, — размер**. Программа Access хочет знать, сколько символов можно сохранить в поле.

Если создать поле типа **Текстовый** и присвоить ему размер 6, то в нем можно будет сохранить имя Joseph, но не Jennifer. Это ограничение часто порождает проблемы. Создавая поле, подумайте о том, какое максимальное количество символов будет храниться в нем, а затем сделайте поле еще немного длиннее. Ничего страшного, если поле будет длиннее, чем нужно (единственный недостаток — оно займет больше места на жестком диске), однако, если поле окажется короче, чем нужно, произойдет ошибка.

- ✓ **Поле МЕМО**. Поле этого типа может содержать до 64 000 символов информации. Это почти 18 страниц книжного текста. Обычно в полях Поле МЕМО хранятся примечания, описания и другие длинные текстовые данные.
- ✓ **Числовой**. Поле содержит число (не забывайте: только одно число). От текста числа отличаются тем, что их можно суммировать, вычитать, умножать, делить. Если несколько цифр записаны в текстовом поле, выполнять над ними эти операции нельзя. Кроме того, если число обозначает количество рублей или долларов, то по многим причинам, о которых мы расскажем далее, лучше применить тип **Денежный**.
- ✓ **Дата/Время**. В зависимости от заданного формата в поле хранится или дата, или время, или оба этих значения. Используется для хранения даты заказа, дней рождения и т.д.
- ✓ **Денежный**. Количество рублей и копеек, долларов и центов, фунтов и шиллингов и т.д. Денежные операции требуют особой аккуратности, поэтому в Access встроены процедуры, заботящиеся о том, чтобы были соблюдены все правила работы с деньгами, например, чтобы в результате округления копеек в чей-либо карман не перетекал денежный ручеек.
- ✓ **Счетчик**. Это же он и делает: считает. При добавлении в таблицу новой записи Access автоматически записывает ее номер в поле Счетчик. Этим обеспечивается уникальность каждой записи. Как вы увидите позже, во многих случаях важно, чтобы каждая запись таблицы чем-либо отличалась от всех других записей. Например, поле Счетчик можно использовать в таблице, содержащей сведения о клиентах. Пользователям базы данных не придется проставлять номера клиентов вручную, это за них автоматически сделает Access.
- ✓ **Логический**. Поле этого типа может содержать одно из двух значений: Да или Нет, истина или ложь, включено или выключено, 0 или 1 — все это разные названия одних и тех же двух значений.
- ✓ **Объект OLE**. Технология OLE (Object Linking and Embedding — связывание и внедрение объектов) получила в последнее время широкое распространение. Объектом OLE может быть что угодно: документ Word, файл изображения, видеофайл, песня, Web-страница. Если объект OLE внедрен в поле таблицы, программа Access увидит, что это такое, и правильно выберет программу для обработки объекта, например, если внедрен видеофайл, то будет открыт видеопроигрыватель.
- ✓ **Гиперссылка**. Благодаря типам полей Access понимает специальный язык гиперссылок, используемых в Интернете. Если база данных Access подключена к сети, поля типа Гиперссылка могут применяться для обращения к другим источникам информации. Более подробно использование гиперссылок в Access рассматривается в главе 8.

Чтобы вы лучше представили себе, как используются столбцы разных типов, в табл. 3.1 приведены примеры столбцов, часто встречающихся в реальных базах данных.

Таблица 3.1. Часто встречающиеся столбцы

Имя столбца	Тип	Размер	Содержимое
Титул	Текстовый	4	лорд, эсквайр, д-р, маркиза, кабальеро
Имя	Текстовый	15	Владимир
Отчество	Текстовый	4	Иванович
фамилия	Текстовый	20	Иванов
Обращение	Текстовый	10	Mrs., мадам, мадемуазель, господин, товарищ, ваше превосходительство
Должность	Текстовый	25	Заведующий складом
Компания	Текстовый	25	Борей Ltd.
Адрес	Текстовый	30	ул. Виноградная, 2
Город	Текстовый	20	Киев
Штат	Текстовый	4	Луизиана
ZIP-код	Текстовый	10	04136 (номер почтового отделения)
Страна	Текстовый	15	Великобритания
Телефон_офиса	Текстовый	12	0959221305
факс	Текстовый	12	0959141499
Домашний_телефон	Текстовый	12	0443331516
Мобильный_телефон	Текстовый	12	380979921506
Электронный_адрес	Текстовый	30	my_name@borej.com
Web-сайт	Гиперссылка	-	www.borej.com
Телекс	Текстовый	11	3529548309843987
SSN	Текстовый	11	45-0-454-599 (номер карточки соцстраха)
Примечания	Поле MEMO	-	Компания поставяляет продукцию со склада в Нижнем Новгороде

Большинство полей, приведенных в качестве примера в табл. 3.1, имеют тип **Текстовый**, даже когда в поле записаны цифры, например номер телефона. Программа Access интерпретирует содержимое такого поля как текст, а не как число. Некоторые упомянутые в таблице типы полей не текстовые, например Поле MEMO и Гиперссылка. Их тоже можно считать текстовыми, но обладающими некоторыми дополнительными характеристиками. Если вас сбивает с толку вопрос, в каких случаях компьютер считает последовательность цифр числом, а в каких — текстом, пока что не думайте об этом. Далее мы расскажем вам о преобразовании чисел в текст и наоборот, о правилах такого преобразования и о том, в каких местах базы данных его нужно применять. Для текстовых полей можно задавать правила форматирования. Например, если в поле записывается номер телефона, значит, оно должно содержать только заданное количество цифр с дефисами в определенных местах. В Access можно задать автоматическую проверку этих условий. Если пользователь ошибется и введет строку, которая не может быть номером телефона, программа сообщит ему об этом.

Задание имен полей

Из всех программ управления базами данных, работающих под управлением Windows, в Access используются наиболее простые правила именования полей. Ниже приведены несколько рекомендаций.

✓ **Начинайте имя поля с буквы.** Программа разрешает начать имя с любого символа, допустимого в именах, но начинать имя все же лучше с буквы по следующим причинам. Во-первых, если шрифт мелкий, разглядеть символ тяжелее, чем букву, поэтому другим людям будет тяжелее работать с базой данных. Во-вторых, имя несет смысловую нагрузку. Назначение поля должно быть ясным с первого взгляда на его имя. Например, если поле содержит адрес, то имя `Адрес` для него наиболее подходящее. Уточняющие символы (если адресов несколько) лучше располагать в конце имени, а не в начале. В имени поля можно использовать пробелы, однако рекомендуется вместо них применять символы подчеркивания. Это обусловлено тем, что во многих программах управления базами данных, в отличие от Access, пробелы в именах запрещены. Если когда-либо в бу-

душем нужно будет экспортировать таблицы, пробелы в именах породят серьезные проблемы. Символы, запрещенные в именах полей Access, приведены в табл. 3.2.

✓ **Имя поля должно быть коротким и понятным.** В Access поддерживаются имена полей длиной до 64 символов, но даже не помышляйте о том, чтобы воспользоваться этой возможностью. С другой стороны, не скупитесь на символы и не применяйте имена вроде `a1` или `Uxc`, которые ничего не говорят о назначении поля.

✓ **Постарайтесь использовать в именах только буквы, цифры и символы подчеркивания.** Имя — это не бижутерия, а рабочий инструмент. Оно не должно быть вычурным. В последнее время стало модно “издеваться” над именами. Видимо, вы заметили, что в имена многих торговых марок в конце добавляют символы “+” или “#”, чередуют строчные и прописные буквы в странной последовательности, а иногда даже ставят точку в качестве первого символа имени. Люди будто соревнуются в экстравагантности. Эта мода ощущается и в базах данных. Не поддавайтесь ей.

Таблица 3.2. Символы, запрещенные в именах полей Access

Символ	Название
/	Прямая черта дроби
*	Звездочка
;	Точка с запятой
:	Двоеточие
!	Восклицательный знак
#	Решетка (шарп, диэз)
&	Амперсанд
?	Вопросительный знак
-	Дефис (минус)
"	Двойная кавычка
'	Одинарная кавычка
\$	Доллар
%	Проценты

Плоские и реляционные базы данных

Два наиболее распространенных типа баз данных — *плоские* и *реляционные*. Существуют и другие типы, но нам они не понадобятся.

Изолированные таблицы

В плоской базе данных вся информация “свалена” в единственную таблицу, никак не связанную с другими таблицами, размещенными в других базах данных. Плоская база данных расположена в *плоском файле*. Характерный пример плоской базы данных — телефонный справочник, содержащий три столбца: фамилии, адреса и номера телефонов. В некоторых местах справочника информация дублируется. Например, если у человека есть три телефона, то его адрес повторяется в трех местах. Однако это не порождает никаких проблем, и база данных работает прекрасно.

Отношения между таблицами

Реляционная база данных содержит несколько таблиц, причем таблицы связаны между собой. При разработке реляционной базы данных обычно стремятся устранить *избыточную* информацию путем разбиения одной таблицы на несколько. Рассмотрим предыдущий пример с телефонным справочником.

Если у клиента телефонной станции есть три телефона, то в плоской базе данных ему посвящены три строки, причем во всех трех строках его фамилия и адрес одни и те же. Чтобы устранить избыточность, создадим две таблицы вместо одной и пронумеруем клиентов. В первой таблице разместим *идентификационный номер* (ИД), фамилию и адрес клиента, а во второй — ИД и номер телефона клиента. Благодаря этому клиент, имеющий три телефона, в первой таблице будет иметь только одну запись, а его фамилия и адрес не будут храниться в трех местах. Во второй таблице тоже не будет дублирования информации.

Ключ

Ключом называется столбец (или, если хотите, поле), связывающий между собой две таблицы реляционной базы данных. Значения ключевых полей, совпадающие в двух таблицах, идентифицируют отношения между записями в разных таблицах.



Значение ключевого поля похоже на ярлычок, прикрепляемый к рубашке в прачечной. Чтобы получить свою рубашку, вы предъявляете ваш экземпляр ярлычка (номер на жетоне или в квитанции), а сотрудник прачечной сопоставляет его с экземпляром ярлычка на рубашке. Если идентификационные номера на ярлычках равны, значит, рубашка ваша. Идентификационный номер — важное и часто используемое понятие баз данных. Для его обозначения даже используется специальное сокращение — ИД (или ID).

Вернемся к примеру с телефонным справочником. Каждый клиент имеет уникальный ИД. Вторая таблица, содержащая телефонные номера, состоит из двух столбцов, в которых хранятся ИД и телефоны клиентов. Чтобы найти, кому принадлежит телефонный номер, записанный во второй таблице, нужно посмотреть на ИД, ассоциированный с телефонным номером, найти в первой таблице запись с этим ИД, а в записи найти фамилию клиента. Естественно, для этого нужно выполнить больше операций, чем в случае, когда телефонный номер и фамилия находятся в одной таблице, как в плоской

базе данных. Однако это почти единственный недостаток реляционных баз данных по сравнению с плоскими. Преимущества же — огромные. В первую очередь — экономия памяти за счет устранения избыточной информации. В нашем простом примере экономия памяти небольшая, но в реальных задачах она может быть в десятки или даже в сотни раз. К тому же, когда информация дублируется, в нее тяжело вносить изменения, поскольку для этого нужно найти все места, в которых она продублирована, а это — чрезвычайно сложная задача.



Если все вышесказанное кажется вам чрезмерно сложным, не огорчайтесь. Реляционные базы данных действительно сложные. Однако все это происходит “за кулисами”. Все, что вам нужно, Access делает для вас автоматически. Access — умная и услужливая программа. Для пользователя достаточно знать, как ввести данные в поля формы, на какой кнопке щелкнуть, чтобы вывести отчет, как просмотреть содержимое таблицы. Если же вы собираетесь разрабатывать базы данных, вам все же придется заглянуть за кулисы, чем мы и займемся более подробно в главе 4. А пока что рассмотрим еще несколько простых операций с Access.

В каких случаях нужно связывать таблицы

Теперь вы понимаете различия между плоской и реляционной базами данных. Нужно ли вам это? Безусловно, нужно. Каждый способ хранения данных обладает своими преимуществами и недостатками.

- ✓ **Плоскую базу данных легче создать и поддерживать.** Характерный пример плоской базы данных — электронная таблица Excel. В ней хранится набор записей. Каждая запись — это строка таблицы. В таблицу можно добавлять сколько угодно строк. Все просто и легко. Однако компания Microsoft в своем стремлении к усовершенствованию Excel добавила в нее довольно много средств, характерных для реляционных баз данных, возможно даже, переусердствовав в этом. В результате, открыв интерфейс Excel, вы, глядя на него, не скажете, что Excel проще, чем Access.
- ✓ **Реляционная база данных предоставляет намного больше возможностей, чем плоская.** Связывание таблиц — незаменимое средство, когда данные хранятся в существенно разных таблицах, например, когда нужно хранить информацию о клиентах, заказах, продуктах, сотрудниках и т.д. Таблицы клиентов и заказов связаны тем, что каждый заказ сделан некоторым клиентом, таблицы заказов и продуктов — тем, что каждый заказ сопровождается определенным сотрудником, таблицы продуктов и заказов — тем, что каждый заказ определяет поставку некоторого продукта. Каким образом все это можно хранить в одной таблице?



Вы можете создать для себя небольшую реляционную базу данных, чтобы хранить в ней, например, сведения о личной библиотеке или о расходах семейного бюджета. Если вы владеете небольшой мастерской, можете создать несложную реляционную базу данных и хранить в ней списки клиентов и прайс-листы комплектующих. Однако не пытайтесь самостоятельно сконструировать базу данных для вашей организации. Это большая и сложная работа. Она затрагивает многих людей и требует изучения существующих бумажных процедур обработки потоков информации. И даже если вы намерены создать базу данных

только для себя, не торопитесь приступать к ее конструированию. Сначала изучите эту книгу более подробно. Чем о большем количестве средств Access вы узнаете, тем более совершенной и полезной будет ваша база данных.



Хотя Access является программой управления реляционными базами данных, с ее помощью несложно создать и плоскую базу данных. Для этого нужно всего лишь создать одну таблицу, создать в ней столбцы и присвоить столбцам имена.

Построение базы данных

Вы прочли несколько первых глав. Возможно, вы пролистали следующие главы, и теперь чувствуете, что готовы сконструировать собственную базу данных. Не забывайте о совете не торопиться, приведенном в предыдущем разделе. В Access есть инструменты, специально предназначенные для построения простых баз данных и носящие гордое имя *мастеров (wizards)*. Однако сначала рассмотрим создание базы данных вручную.

Выполнив приведенные ниже действия, вы создадите новую базу данных и добавите в нее таблицу.

1. Если Access в данный момент не открыта, запустите ее.

Запуск Access рассматривается в главе 2.

В рабочей области вы увидите пиктограмму Новая база данных (New Blank Database).

2. Щелкните на пиктограмме Новая база данных.

Возможно, пиктограмма Новая база данных на вашем экране не появилась. Это объясняется тем, что разрешение экрана слишком низкое, и для нее не нашлось места. Тогда увеличьте разрешение экрана (см. главу 2) или откройте меню быстрого доступа и выберите команду Создать (Create). В любом случае справа будет активизирована панель Новая база данных (рис. 3.1).

3. Введите имя базы данных, заменив предлагаемое по умолчанию.

Расширение .accdb вводить не нужно. Если его нет, Access добавит его автоматически.

Под текстовым полем с именем базы данных приведен маршрут текущего каталога, в котором она будет сохранена при щелчке на кнопке Создать (Create).

4. Если хотите сохранить новую базу данных в другом каталоге, щелкните на пиктограмме, расположенной справа от имени файла, и выберите нужный каталог.

При щелчке на пиктограмме активизируется диалоговое окно Файл новой базы данных (File New Database), показанное на рис. 3.2. В нем найдите и откройте нужный каталог. Он может быть расположен на вашем жестком диске или в сети, если компьютер подключен к локальной сети. Определив место для новой базы данных, щелкните на кнопке ОК, чтобы закрыть диалоговое окно Файл новой базы данных.

5. Щелкните на кнопке Создать.

Будет создана новая база данных и открыта пустая таблица с незамысловатым именем Таблица1 (рис. 3.3). На левой панели приведен список существующих таблиц, состоящий пока что из единственного пункта.



После щелчка на кнопке Создать может появиться диалоговое окно, спрашивающее, заменить ли существующий файл базы данных. Это означает, что файл с таким именем уже существует в заданном каталоге. Если файлы с этим именем есть в других каталогах, диалоговое окно не появится, и Access позволит вам создать файл.

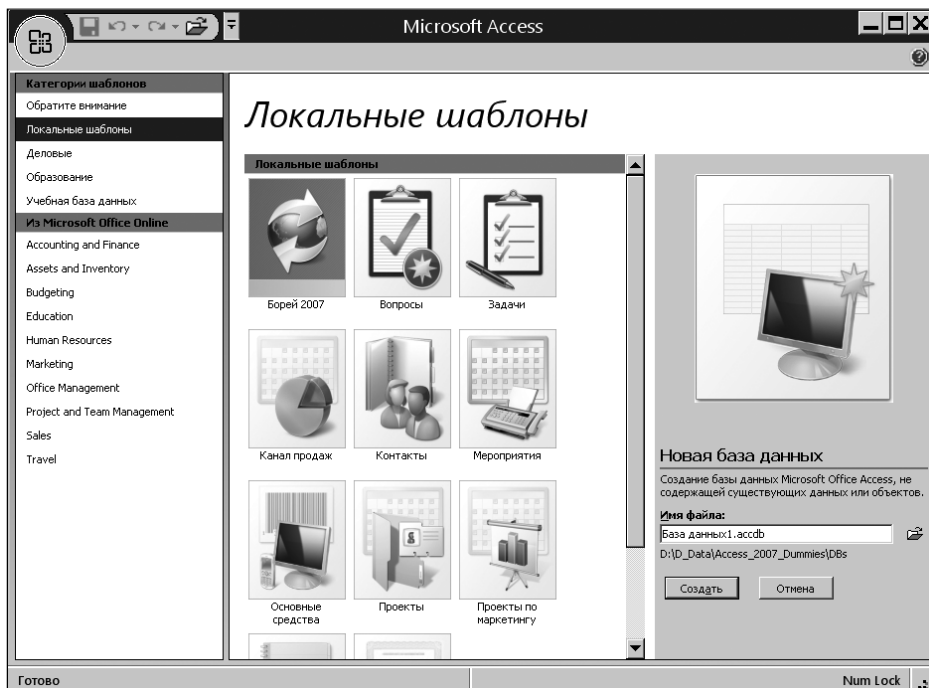


Рис. 3.1. Задайте имя и расположение новой базы данных

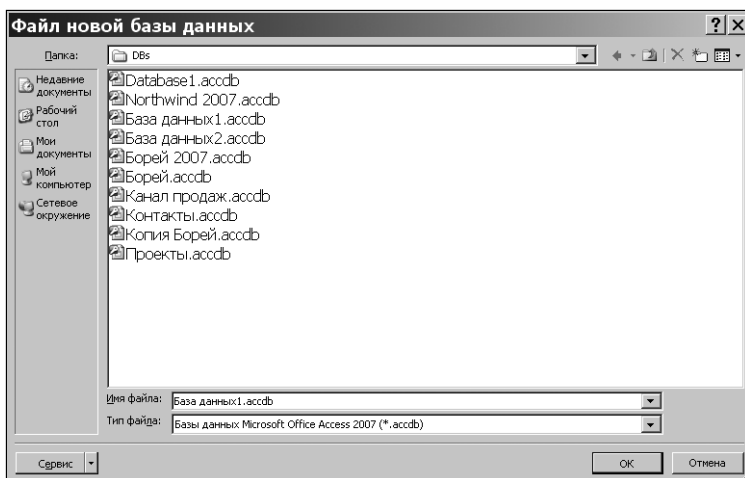


Рис. 3.2. Выбор расположения новой базы данных

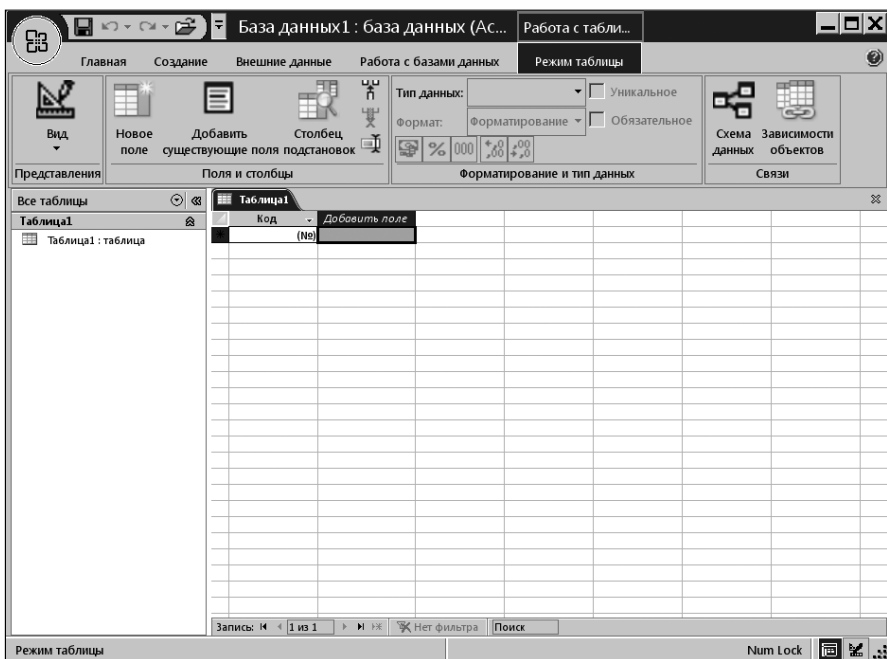


Рис. 3.3. Новая таблица новой базы данных: все чистое, как в первый момент сотворения мира

- Щелкнув на кнопке Нет, вы можете вернуться к предыдущему диалоговому окну, чтобы изменить имя базы данных (напоминаем, что “имя базы данных” и “имя файла базы данных” — это одно и то же).
 - Если старый файл вам не нужен, щелкните на кнопке Да. Он будет удален, а вместо него в каталог будет записан файл создаваемой вами базы данных.
6. Чтобы добавить столбец, дважды щелкните на надписи **Добавить поле** (не ухмыляйтесь!), расположенной в заголовке таблицы. Введите имя нового столбца (ну ладно, поля) и нажмите клавишу <Enter>.



Столбец Код (ID) уже существует в таблице. Если он не нужен, можете удалить его. Однако выше в главе мы уже неоднократно говорили о его важности. В него Access автоматически добавляет идентификационные номера (ИД) записей таблицы при создании новых записей. Этим обеспечивается уникальность каждой записи. Можете изменить имя столбца ИД, например, на Номер_клиента. Для этого дважды щелкните на имени, введите новое имя и нажмите клавишу <Enter>.

Позже, когда вы создадите другие таблицы и начнете устанавливать между ними отношения, столбец ИД будет играть роль *первичного ключа*.

7. Создайте еще один столбец.

После ввода имени предыдущего столбца вы нажали клавишу <Enter>. В этот момент появляется следующий столбец, а курсор ввода устанавливается в его заголовок (рис. 3.4).

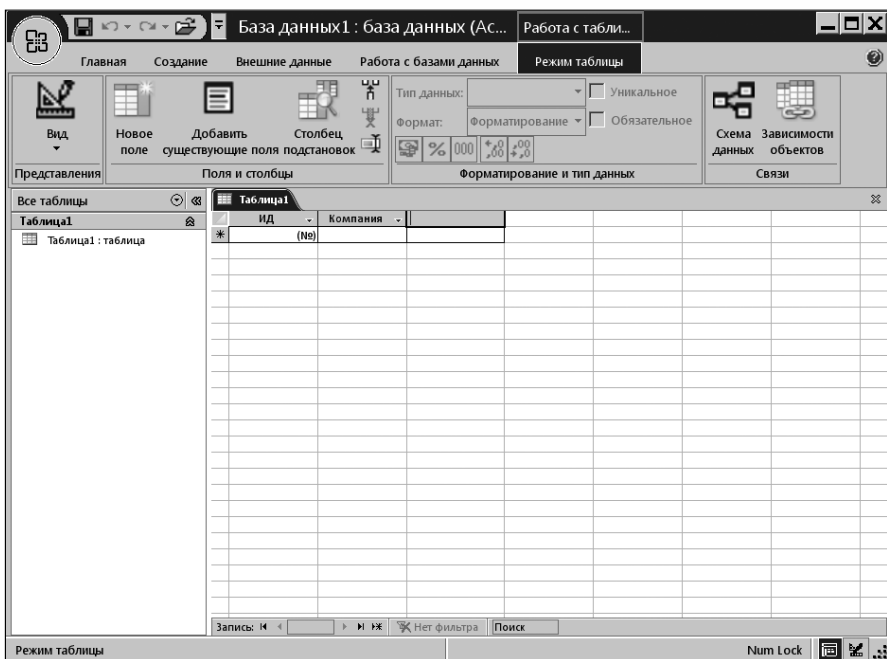


Рис. 3.4. Курсор ввода находится в заголовке нового столбца; можете вводить имя столбца

Повторяя пп.6-7, создайте в таблице все необходимые столбцы. Пока что не очень беспокойтесь об их именах, изменить имя сейчас можно в любой момент. Позже, когда вы создадите формы, отчеты и запросы, в них будет упоминаться имя столбца, поэтому изменить его будет сложнее.

8. Чтобы сохранить новую таблицу и всю базу данных, нажмите клавиши <Ctrl+S> или щелкните на кнопке **Сохранить (Save), расположенной на панели быстрого доступа.**

Команду Сохранить рекомендуется запускать после каждой важной операции. Иначе, если случится что-либо непредвиденное (крах операционной системы, критическая ошибка, отключение питания), вам тяжело будет вспомнить, что вы делали, и восстановить базу данных.



Имя таблицы Таблица1 вряд ли сообщает что-либо полезное вам или другим людям, которые, возможно, будут работать с вашей базой данных. Имя таблицы легко изменить до или после сохранения базы данных. Для этого выполните следующие действия.

1. Щелкните правой кнопкой мыши на корешке вкладки таблицы (т.е. на ее имени).
2. В контекстном меню выберите команду **Сохранить**. Учитывайте, что сохраняется имя таблицы, а не вся база данных.
3. В появившемся диалоговом окне **Сохранение** введите новое имя.
4. Щелкните на кнопке **ОК** или нажмите клавишу <Enter>.
5. Сохраните базу данных, чтобы зафиксировать изменения.

Добавление и удаление таблиц

Пока что никто не требует совершенства от вашей базы данных. По крайней мере, при первом вашем погружении в пучины таблиц. И даже при втором и третьем. Даже опытные специалисты по базам данных часто ошибаются или забывают сделать что-либо. Это вполне обычная ситуация. Построив базу данных, человек видит, что какая-то таблица не нужна, или, наоборот, что он забыл создать таблицу, нужную для запроса или отчета.

Как же быть в таком случае? Никаких проблем! С помощью графического интерфейса Access удалить или создать таблицу можно, несколько раз щелкнув кнопкой мыши.

Добавление таблицы

Поскольку в реляционной базе данных используется больше одной таблицы, вам нужно уметь добавлять их.

Чтобы добавить новую таблицу в существующую базу данных, выполните следующие действия.

1. Щелкните на вкладке **Создание (Create)**, чтобы активизировать ее.

На экране появятся инструменты вкладки Создание (рис. 3.5).

2. На ленте щелкните на кнопке **Таблица (Table)**.

Будет создана и открыта новая пустая таблица с именем Таблица2 (рис. 3.6). В этот момент можно вводить имя первого поля.

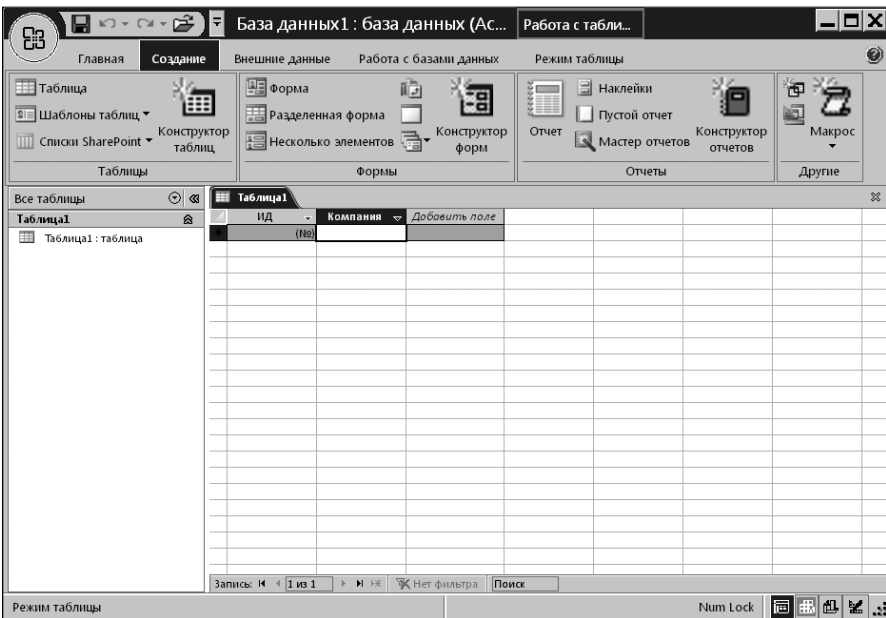


Рис. 3.5. Вкладка Создание содержит инструменты создания новых компонентов базы данных

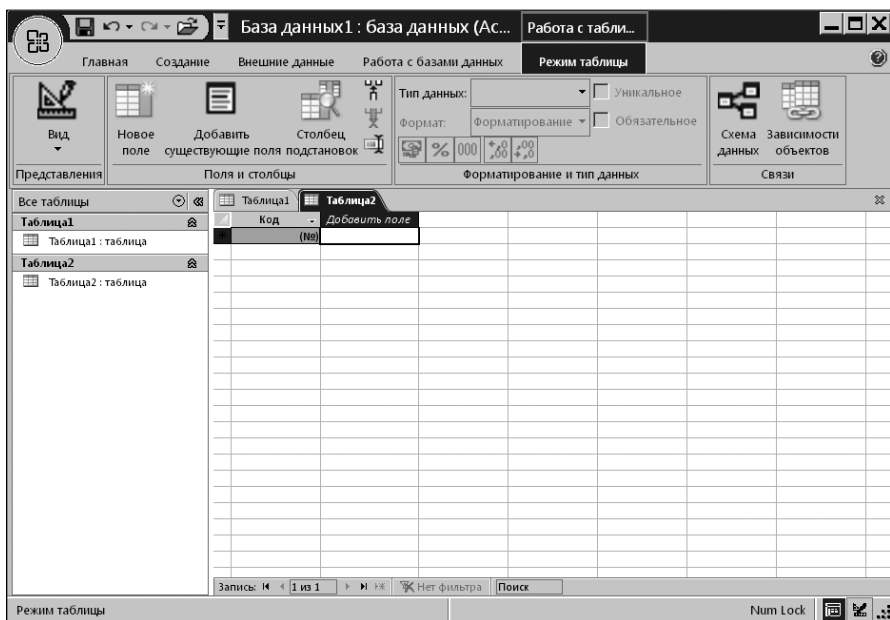


Рис. 3.6. Новая пустая таблица

3. Добавьте в новую таблицу столбцы и присвойте им имена, как рассказано в предыдущем упражнении.



Во время работы над базой данных почаще сохраняйте ее.

4. Повторяя пп.1–3, создайте все необходимые таблицы.



Можете не выполнять эту операцию непосредственно сейчас. В базу данных Access легко в любой момент добавлять таблицы, переименовывать их, добавлять и удалять столбцы и т.д. Цель упражнения — не создать совершенную базу данных, а научиться создавать таблицы.

Удаление таблицы

Предположим, создав таблицу, вы обнаружили, что она вам не нужна. Возможно, после создания таблицы С оказалось, что нужны только таблицы А и В или что не нужна таблица D, созданная после таблицы С. В любом случае таблицу легко удалить, даже если в ней уже есть записи.



Удалить таблицу легко. Может быть, даже слишком легко. Однако перед удалением тщательно убедитесь в том, что она действительно не нужна, иначе вы можете безвозвратно потерять ценную информацию или разорвать связи, восстановить которые будет тяжело. При удалении таблицы мгновенно удаляются все ее связи с другими таблицами и с отчетами, формами, запросами. Впрочем, перед удалением Access выведет диалоговое окно, предлагающее еще раз хорошенько подумать, прежде чем выполнить опасную операцию.

Чтобы удалить таблицу, выполните следующие действия.

1. Когда база данных открыта, взгляните на левую панель.

На ней приведен список всех таблиц, каждая из которых представлена длинной горизонтальной кнопкой (рис. 3.7). Закройте удаляемую таблицу, щелкнув на пиктограмме ×, расположенной в правом конце строки корешков. Открытую таблицу удалить невозможно.

2. Щелкните на паре стрелочек, направленных вниз и расположенных справа от имени таблицы, которую нужно удалить.

Имя таблицы будет продублировано в строке, появившейся под исходной кнопкой.

3. Щелкните правой кнопкой мыши на имени таблицы в этой новой строке и выберите команду Удалить (Delete), как показано на рис. 3.8.

4. Щелкните на кнопке Да, чтобы подтвердить удаление таблицы.

Таблица будет удалена.

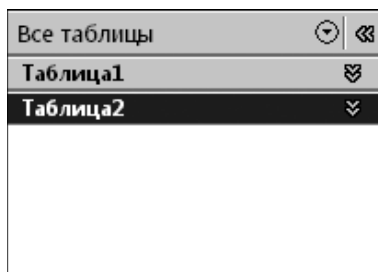


Рис. 3.7. Каждая таблица базы данных представлена длинной кнопкой

Сейчас вы, наверное, думаете, что наконец-то настало время заполнить таблицы данными. Потерпите еще немного. Прежде чем заполнять таблицы, нужно создать между ними связи, установить ключевые поля, которые свяжут таблицы, и определить параметры каждого столбца, в частности, типы столбцов, рассмотренных выше в главе.

Перед вводом данных нужно хорошенько продумать и зафиксировать параметры столбцов (очень желательно зафиксировать навсегда, хоть это и редко получается). Изменить параметры столбца до его заполнения несложно, Однако, когда столбец содержит сотни заполненных полей, изменение его параметров может оказаться невыполнимой задачей. Вам придется заново вводить в него все данные.



В главе 4 мы расскажем, как подготовить базу данных к выполнению реляционных обязанностей. В главах 5-6 рассматривается настройка столбцов. И только после этого вы начнете вводить данные в таблицы и пользоваться великолепными возможностями, предоставляемыми программой Access.

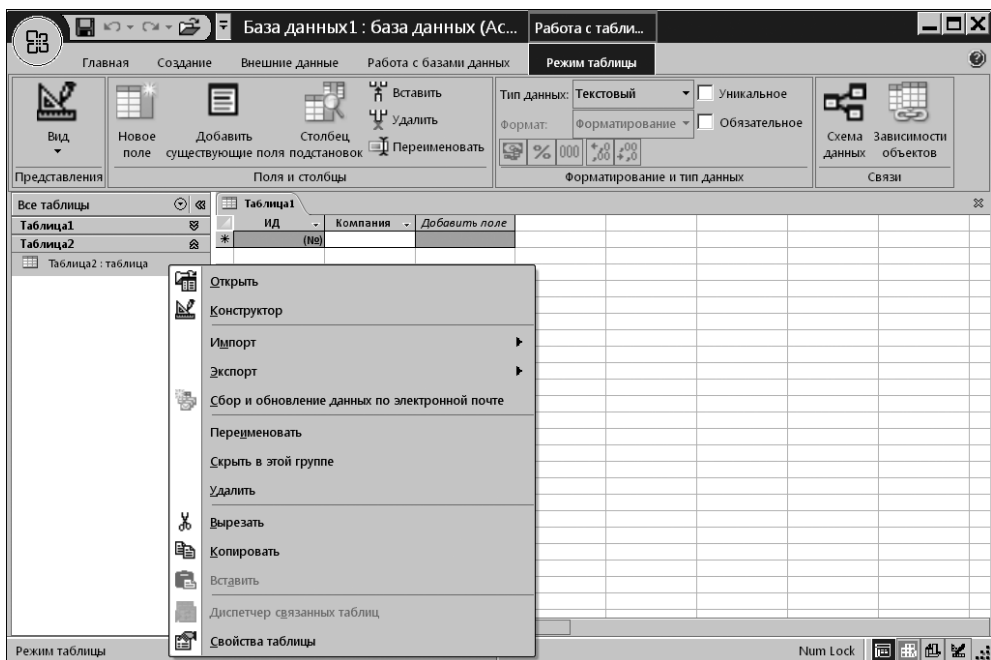


Рис. 3.8. Для удаления таблицы выберите команду Удалить