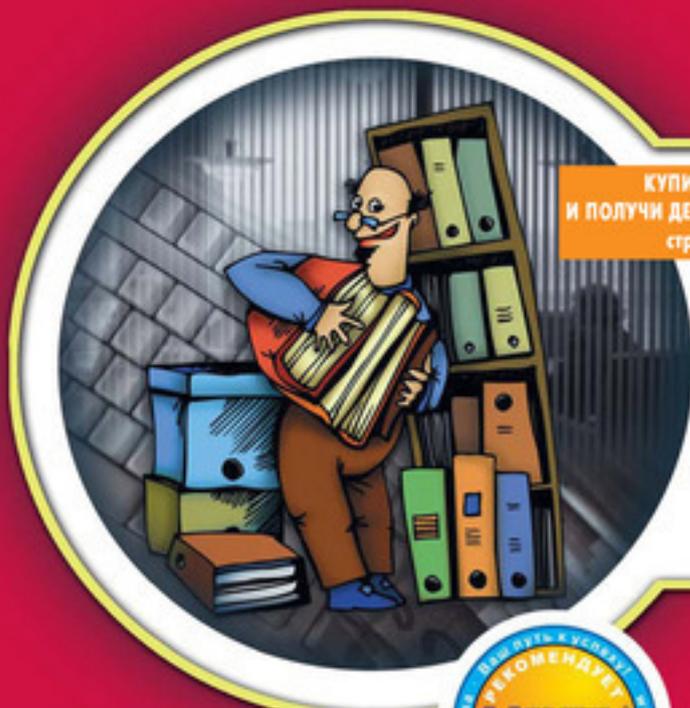


ДУБНОВ П. Ю.

ACCESS 2002



КУПИ КНИГУ
И ПОЛУЧИ ДЕНЬГИ ОБРАТНО!
стр. 320

Центр Компьютерного Обучения
при МГТУ им. Н.Э. Баумана
Рекомендует!



АВТОРИЗОВАННЫЙ ЦЕНТР



САМОУЧИТЕЛЬ

Павел Дубнов

Access 2002: Самоучитель

«ДМК Пресс»

2002

Дубнов П. Ю.

Access 2002: Самоучитель / П. Ю. Дубнов — «ДМК Пресс», 2002

В книге рассматривается широкий круг вопросов, связанных с использованием программной среды Access 2002, которая является составной частью пакета Office 2002 и предназначена для создания банка данных в самых различных предметных областях. Подробно описывается методика проектирования объектов базы данных (таблицы, формы, отчеты, страницы доступа к данным, запросы, модули). Детально обсуждаются вопросы создания интегрированной базы данных в единой среде Access 2002: формирование БД с нуля, конвертирование в программную среду баз данных, созданных в ином программном окружении – Clarion, FoxPro. Особое внимание уделяется формированию разнообразных запросов к интегрированной базе данных Access 2002 с использованием языков программирования SQL, VBA и макросов. Приводятся общие сведения о возможностях языка обмена данными между различными компьютерами и приложениями (XML). Описываются возможности использования гиперссылок, связывающих базу данных с другими программными продуктами. Объясняется, как можно работать с базой данных Access 2002 без установки ее на компьютер, используя технологию ODBC (Open Data Base Connectivity). В приложениях приводятся количественные параметры Access 2002 и связанная с этой СУБД терминология. Предлагаемая книга будет полезна специалистам, занимающимся практической разработкой банков данных и приложений на их основе, а также студентам вузов, изучающим информатику.

© Дубнов П. Ю., 2002

© ДМК Пресс, 2002

Содержание

Об авторе	6
Введение	7
Глава 1	9
Режимы функционирования банка данных в производственных условиях	10
Пользовательские запросы к банку данных	11
Проблемы, связанные с выбором СУБД	11
Вопросы, рассмотренные в настоящей книге	13
Резюме	14
Глава 2	15
Состав информации	16
Что понимать под структуризацией информации	17
Показатели	17
Необходимость структуризации	18
Технология структуризации	19
Пример структуризации данных	20
Проектирование логической структуры базы данных	22
Распределение полей по файлам	22
Файлы и связи между ними	23
Резюме	24
Глава 3	25
Варианты создания таблиц	26
Формирование таблицы в режиме ввода	28
Ввод данных	28
Создание таблицы в режиме конструктора	31
Конец ознакомительного фрагмента.	32

Павел Юрьевич Дубнов

Access 2002

Об авторе

Дубнов Павел Юрьевич – кандидат технических наук, доцент кафедры математики и информатики Института международных экономических отношений (ИМЭО), г. Химки. Автор более 60 научных работ.

Область научных интересов – информатика и информационные технологии, автоматизированные системы управления и информационные системы, разработка математических моделей, статистическая обработка информации.

Введение

Автоматизированные банки данных уже давно стали неотъемлемой частью практически всех компьютерных систем управления на любом уровне – от отрасли до отдельного предприятия.

Однако проектирование и создание баз данных (БД) до сих пор остается, за редким исключением, не технической задачей, а творческим процессом, который скорее сродни искусству, нежели науке. Это утверждение может показаться несколько странным, поскольку разработка и исследование баз данных ведутся более 35 лет. Однако, как нам кажется, такой парадокс вполне объясним. За прошедшие годы неизмеримо вырос уровень потребительских качеств систем управления базами данных (СУБД): разнообразие поддерживаемых функций, удобный для пользователя интерфейс, сопряжение с программными продуктами, в частности с другими СУБД, возможности для работы в сети и т. д.

Но изменения почти не коснулись того, что раньше называлось логическими структурами баз данных. Это формы, в которых пользователь представляет и хранит свою информацию в БД. А ведь именно от них в немалой степени зависит удобство работы пользователя с базой данных: формулировка запроса, простота поиска данных, форма выдачи итоговой информации и другие операции. В современных БД могут использоваться более или менее удачные структуры, но почти никогда мы не найдем обоснованного ответа на вопрос, почему для конкретной базы данных была выбрана именно такая форма.

Однако к настоящему времени накоплен значительный опыт проектирования банков данных, предназначенных для управления производством. Это позволяет сделать процесс создания БД значительно более формализованным. (Правда, поле для субъективных решений, а значит, и для индивидуального творчества все равно остается, но его можно существенно сузить.)

Итак, речь идет об информации, которая формируется и накапливается в компьютерных банках данных. В условиях реальной производственной деятельности это понятие употребляется в двух различных значениях:

- информация, прежде хранившаяся на бумажных носителях и внесенная в новый банк данных, который создавался на основе какой-либо СУБД. Сюда же следует отнести и сведения, связанные с текущим производственным процессом. Они вводятся в банк данных в реальном масштабе времени;
- банк данных, который был создан ранее и используется до сих пор.

Постепенно разница между двумя названными типами данных стирается. С одной стороны, неизбежно появляется новая информация, которую надо структурировать и организовать в банке данных, и создаются новые СУБД, более удобные, чем прежние. С другой стороны, ранее накопленные сведения продолжают храниться в банке данных, который наверняка никто никогда не будет перестраивать. Обычно самое простое решение проблемы – конвертировать старые данные в новую СУБД, объединяя информационные массивы и решая возникающие при этом проблемы. В результате возникает новый банк данных, куда входят разные БД¹. Все они имеют один формат данных (например, для Access – это. mdb), но сохраняют прежнюю

¹ В современной научной литературе понятия «база данных» и «банк данных» часто смешиваются. Традиционно эти термины определялись следующим образом. База данных понималась как набор связанных таблиц, запросов, форм, отчетов, макросов и модулей. Так, в среде Access база данных представляет собой единый файл с расширением. mdb. Банк данных рассматривался как совокупность различных баз данных и программ по их обслуживанию. Однако если база данных включает такие объекты, как форма, отчет, запрос, модуль и др., то в нее входят также программы создания и обслуживания объектов. Тогда под банком данных следует, видимо, понимать множество баз данных в едином формате (скажем, в том же. mdb) и набор программ, позволяющих работать с этими данными.

структуру первичных файлов, таблиц и т. д. Иными словами, данные остаются в значительной мере разнородными, что осложняет дальнейшую работу с ними.

Поэтому после конвертации или интеграции разнородных баз данных в единую программную среду пользователю потребуется дополнительное (иногда специальное) программное обеспечение, чтобы обслуживать полученную БД.

До сих пор в специальной литературе рассматривалась только одна сторона задачи – создание новых баз данных. О том, как конвертировать собранную информацию в программную среду новой СУБД, написано значительно меньше. Наконец, мало внимания уделялось другой важной проблеме: какой должна быть единая программная среда и почему.

Настоящее издание призвано в какой-то степени ответить на перечисленные вопросы. Основная цель, которую ставил перед собой автор, заключается в практической направленности книги, рассмотрении некоторых часто встречающихся в практике работы ситуаций. В связи с этим сознательно оставлены в тени отдельные теоретические проблемы СУБД вообще и Access в частности. Этим обусловлена и структура книги:

- в главе 1 проблемы, анализируемые в книге, подробно рассматриваются на конкретных примерах. Кроме того, в данной главе объясняется, почему для решения стоящих перед пользователем проблем целесообразно использовать именно программную среду Access 2002;

- глава 2 посвящена методическим аспектам вопроса – разработке такой структуры данных, которая не будет зависеть от программной среды;

- в главах 3–6 рассматривается создание базы данных в программной среде Access 2002. Поскольку эта среда содержит элементы, которых не было в предыдущих версиях, приводятся необходимые пояснения;

- в главе 7 изучаются вопросы, связанные с конвертированием баз данных из других СУБД в Access 2002;

- глава 8 содержит общие сведения о проектах Microsoft Access 2002. С введением этой категории Access превращается в распределенную систему.

- в главах 9-12 рассматривается программирование в объединенном банке данных и его различные варианты: программирование на языках SQL и Visual Basic, а также с использованием макросов.

Соглашения

Структура книги подразумевает, что читатель может изучать ее последовательно или использовать как справочник. Небесполезно учесть следующее:

- *курсивом* выделяются термины, встретившиеся впервые;

- **полужирным шрифтом** отмечены элементы интерфейса операционной системы и рассматриваемых программ (пункты меню, заголовки диалоговых окон и пр.), а также клавиши;

- последовательно выполняемые команды меню записываются через стрелочку, например: **Файл → Открыть**;

- при обозначении сочетаний клавиш, которые следует нажимать одновременно, используется знак «плюс», например: **Ctrl+O**;

- моноширинным шрифтом отмечены коды программ;

- все Internet-адреса подчеркнуты.

Таким образом выделена особо важная информация, примечания, предупреждения.

Глава 1

Постановка проблемы

Сегодня никого нельзя удивить понятием «компьютерный автоматизированный банк данных». Это так же обычно, как передача сообщений по факсу или заказ билетов по телефону. Существует бесчисленное множество функционирующих банков данных, однако далеко не все они удобны в эксплуатации. Пользователю, работающему с банком данных, необходимо «умение» последнего быстро, гибко и в требуемой форме организовать и выдать информацию в ответ на любой запрос.

В этой книге подробно рассказывается обо всех проблемах, с которыми может встретиться разработчик баз и банков данных: начиная с логической структуризации и заканчивая написанием сложных макрокоманд. Особое внимание уделено вопросам интеграции разнородных баз данных в единый эффективно функционирующий информационный банк. Все решения показаны на примере реальной базы данных по чрезвычайным ситуациям, о содержимом и структуре которой рассказано в начале главы 2.

Режимы функционирования банка данных в производственных условиях

Как правило, имеются в виду следующие режимы функционирования банка данных:

- режим начальной загрузки, в котором исходная информация, содержащаяся в банке данных, вводится в соответствующие структуры БД;
- режим корректировки, в котором осуществляется обновление, добавление и удаление информации, находящейся в банке данных;
- режим диалога, в котором пользователи обращаются к банку данных и производится обработка запросов. Такие запросы могут предусматривать:
 - только выдачу пользователю информации о тех или иных параметрах процесса. Эта информация в требуемом формате содержится в банке данных;
 - решение поставленной задачи с использованием сведений, находящихся в банке данных;
- режим реорганизации и анализа, в котором выполняются операции, непосредственно связанные с поддержанием банка данных в рабочем состоянии:
 - реорганизация структур БД;
 - копирование и восстановление БД;
 - анализ статистических данных, связанных с функционированием информационного фонда.

Пользовательские запросы к банку данных

Из всех перечисленных выше запросов для пользователя наиболее важен режим диалога, а остальные носят служебный, вспомогательный характер. Режим диалога позволяет формировать самые различные запросы и является первым и необходимым шагом к аналитической обработке информации.

Конечно, нельзя заранее предусмотреть все возможные варианты запросов. Назовем лишь самые характерные типы запросов пользователя к банку данных в порядке возрастания сложности:

- запросы на обработку данных, связанных с одной таблицей (выборка, удаление, корректировка и ввод данных);
- запросы на групповую обработку данных (сумма, среднее значение и т. д.), связанных с одной таблицей;
- запросы, при которых условием отбора записи является полное значение поля;
- запросы, при которых условием отбора записи является неполное значение поля;
- запросы с несколькими условиями отбора записей в разных полях;
- запросы с несколькими условиями отбора записей в одном поле;
- запросы с заданием параметров;
- запросы на создание объединенной выборки из нескольких разнородных таблиц и т. д.

Как уже говорилось, наличие в банке разнородных баз данных несколько усложняет работу. Подробнее этот вопрос рассматривается в следующих разделах.

Проблемы, связанные с выбором СУБД

Говоря о создании банка данных и его последующей работе в производственном режиме, надо определить, в какой программной среде он будет функционировать. Вопрос этот не так прост, и при его решении надо учитывать два существенных аспекта проблемы:

- для упорядоченного накопления и хранения поступающей информации приходится разрабатывать новые базы данных. Логично, что вы будете оценивать различные СУБД именно с этой точки зрения. Здесь не требуется особых комментариев;

- история использования компьютерных банков данных в СССР и в постсоветской России насчитывает более 35 лет, за этот период сменилось несколько поколений СУБД. Можно увлеченно спорить о том, насколько рациональным был этот процесс и какова эффективность той или иной конкретной СУБД. Однако важнее другое – хороши или плохи были эти системы, но в них аккумулировано значительное количество информации, которая используется в практических целях. Ясно, что с каждым годом объем таких данных возрастает.

Системы управления непрерывно совершенствуются. Мировой опыт показывает, что поколения СУБД сменяются примерно каждые 5 лет. Естественно, все более актуальным становится вопрос конвертирования данных, то есть перевода их в новую программную среду без потери информации. Решая, какую СУБД выбрать, обязательно учитывайте ее возможности конвертирования; они не менее важны, чем удобство разработки БД в данной программной среде.

Названным условиям удовлетворяет СУБД Access, входящая в состав комплексного программного продукта Microsoft Office. Последней версией этой системы на сегодняшний день является Access 2002, или Access XP. Именно эта версия рассматривается в настоящем издании в качестве основной. Поэтому в дальнейшем мы не будем указывать индекс рядом с названием, говоря просто Access. Если же подразумевается какая-либо другая версия, например Access 97 или Access 2000, то она указывается с соответствующим индексом. Правда, на совре-

менном рынке много и других программных продуктов, успешно используемых в качестве платформы для банка данных. Поэтому сформулируем те критерии, на основании которых следует выбирать СУБД, и оценки Access по этим показателям²:

- количество ключевых (дескрипторных) полей, поддерживаемых в СУБД.

В Access ограничения на эту величину отсутствуют;

- ограничение на длину поля.

В Access данное ограничение составляет 255 байт для текстовых полей и до 255 байт – для числовых, в зависимости от типа поля;

- разнообразие типов обрабатываемых полей.

В Access имеются поля, содержащие текстовый и числовой типы данных. Эти типы, в свою очередь, представлены разными вариантами;

- дизайнерские возможности системы.

Наличие в Access мастеров и конструкторов позволяет достаточно быстро создавать таблицы, формы, отчеты, запросы. Добавление диаграмм в формы и отчеты, быстрая настройка программы и анализ ее быстродействия, использование архивариуса, возможность импорта и экспорта файлов, работа с гиперссылками и применение технологии OLE внутри пакета Microsoft Office;

- требования к уровню подготовки проектировщика и пользователя БД. Минимальные. Некоторые программные навыки нужны лишь в том случае, если придется использовать Visual Basic;

- язык программирования, операционная среда, сетевые возможности, требуемые ресурсы.

Язык запросов SQL, Visual Basic, операционная система Windows 98 или выше, а также Windows NT. При полной установке потребуется 26–30 Мбайт оперативной памяти и около 120 Мбайт памяти на жестком диске. Однако Access, как правило, устанавливается в составе программного комплекса Microsoft Office, и тогда требуется около 200 Мбайт памяти на жестком диске. Система Access обладает всеми современными сетевыми возможностями;

- язык представления данных, обработка символьной информации.

Имеются;

- поддерживаемые структуры и форматы данных.

В Access поддерживаются реляционные структуры данных;

- простота освоения системы, наличие русской версии документации.

Первичное освоение займет всего несколько дней. Имеется русифицированная версия Access в составе пакета Microsoft Office, а также русифицированная документация для пользователей различных уровней подготовки;

- поддерживаемый системой математический аппарат.

В Access он достаточно развит и включает операторы, функции, логические выражения и т. д. Более того, статус Access как органической составной части Microsoft Office позволяет легко пользоваться математическим аппаратом Excel;

- поддерживаемые системой возможности обработки и представления графической информации.

В Access поддерживаются операции с диаграммами. Поскольку эта СУБД встроена в пакет Microsoft Office, то пользователь может работать и с другими графическими объектами, входящими в состав данного пакета;

- возможности взаимодействия с другими пакетами прикладных программ (текстовыми редакторами, электронными таблицами, геоинформационными системами (ГИС) и др.).

В рамках пакета Microsoft Office можно работать с Word и Excel;

² Подробные количественные оценки Access приведены в приложении 1.

- возможности корректировки файлов, содержащих данные. В Access это очень просто сделать;
- наличие русифицированной и достаточно подробной справочной системы, а также файлов **Help** (Помощь).

Такая справочная система есть, и она доступна из любого режима в любой момент;

- разнообразие и гибкость формируемых запросов на предоставление данных.

Система Access отвечает этому условию.

Наверное, приведенные выше оценки не дают оснований утверждать, что Access – идеальная СУБД. Однако безупречных СУБД вообще не существует. Сравним, к примеру, Access с такой системой, как Oracle. Последняя – СУБД гораздо более высокого класса, значительно превосходящая Access по своим возможностям. Но такие преимущества имеют и обратную сторону: Oracle громоздка, сложна в освоении и требует для нормального функционирования специальные и очень мощные технические средства. Область применения Oracle – создание очень больших централизованных информационных систем. По-видимому, время их массового использования в России еще не наступило.

В то же время Access является весьма гибкой и универсальной системой, предъявляющей достаточно умеренные требования к техническому обеспечению. Поэтому на сегодняшнем этапе эта СУБД удобна для работы практически на всех иерархических уровнях управления производством – от отрасли в целом до отдельного предприятия.

Вопросы, рассмотренные в настоящей книге

Приведенные выше оценки относятся к версии Access 97. Как уже было сказано, на рынке появилась очередная версия этой СУБД – Access 2002 (Access XP), обладающая более широкими возможностями. Поэтому при изложении материала автор учитывал:

- наличие в версии Access 2002 новых элементов по сравнению с Access 97;
- особенности, обусловленные использованием разнородных баз данных.

В той или иной степени специфика разных версий Access проявляется на всех этапах работы: от создания первичных таблиц до формирования запросов и использования элементов программирования. Можно было или сосредоточиться на том, чем отличаются друг от друга два варианта баз данных и программ (тогда материал неизбежно был бы изложен отрывочно и непоследовательно), или рассматривать весь процесс создания и использования банка данных от начала до конца в каждой из версий, по ходу описания комментируя различия между ними. Хотя во втором случае неизбежны повторы и в какой-то мере дублирование уже имеющейся литературы, для читателя такой вариант удобнее. Все необходимые сведения приводятся в одной книге, и пользователю не придется, забыв какую-то мелочь, «буксовать» из-за этого в повседневной работе.

Чтобы пользователю было легче работать с этой книгой, материал изложен следующим образом.

Все методические рекомендации по структуризации показателей и проектированию логических структур БД применимы к любой версии Access.

В книге описывается процесс создания новых баз данных в программной среде Access 2002. Отличия этой версии от предыдущих версий специально оговариваются.

Что касается конвертирования БД, созданных в других программных средах, то сначала речь пойдет о «переводе» в Access 97, а затем в Access 2002. Кроме того, будут подробно рассмотрены те дополнительные возможности конвертации, которые появились в Access 2002 по сравнению с Access 97.

Резюме

1. Параллельно с разработкой баз данных в новых современных СУБД используется множество банков данных, построенных на основе программного аппарата морально и технически устаревших СУБД. Накопленная в них информация представляет большую ценность, но разрабатывать для этой информации новые базы данных никто никогда не будет. Поэтому очень важной становится проблема конвертации данных с тем, чтобы они обрабатывались совместно с новыми СУБД в рамках единого банка данных.

2. В качестве базовой СУБД для интеграции разнородных СУБД в такой банк данных на сегодняшнем этапе предлагается Access 2002 (Access XP).

3. Целью настоящей книги является обсуждение методических и практических вопросов, связанных с разработкой интегрированного банка данных.

Глава 2

Предпроектная структуризация информации

Эффективность работы банка данных во многом зависит от того, как структурирована накапливаемая в нем информация. В этой главе как раз и говорится обо всех проблемах, связанных с определением логической структуры данных.

В настоящей книге будут рассматриваться в основном примеры из определенной *предметной области* – тематической сферы, к которой относится обрабатываемая информация. Речь пойдет о чрезвычайных ситуациях (ЧС), происходивших в действительности; о работах, связанных с ликвидацией последствий ЧС, и, в частности, об используемых при этом контрольно-измерительных приборах. Автор опирался на информацию, которая содержится в банках данных Министерства РФ по делам гражданской обороны, чрезвычайных ситуаций и ликвидации последствий стихийных бедствий (впоследствии – Министерства природных ресурсов России), бывшего Госкомитета РФ по охране окружающей среды (Госкомэкологии России) и бывшего Федерального агентства правительственной связи и информации (ФАПСИ). Создание объединенного банка таких данных не завершено, и состав включаемых в него БД в дальнейшем должен расширяться. Полученная информация используется преимущественно в аналитических целях: сбор статистических сведений, выявление тенденций, оценка последствий ЧС, выработка рекомендаций по их предотвращению и т. д.

Состав информации

Наиболее динамичной частью информации, на примере которой рассказывается о возможности Access 2002, являются данные о различных чрезвычайных ситуациях. Прежде всего это:

- непосредственные сведения о ЧС (вид ЧС, дата и место происшествия, объект, на котором произошла катастрофа);
- характеристика ЧС;
- количество пострадавших, в том числе погибших;
- предварительные оценки материального ущерба в стоимостном и натуральном выражении;
- влияние ЧС на жизнедеятельность местного населения, на окружающую среду и функционирование отраслей народного хозяйства;
- возможность или невозможность ликвидации последствий ЧС на месте, ориентировочные сроки такой ликвидации;
- типы и количество единиц оборудования, число специалистов, необходимых для ликвидации последствий ЧС;
- характер и примерные объемы выполняемых работ.

Менее динамичная часть информации – данные о контрольно-измерительных приборах, которые используются при ликвидации последствий ЧС.

Постоянная часть информации – словари понятий, встречающихся в банке данных.

Описываемый банк данных состоит из следующих разделов:

- база данных, разрабатываемая в среде СУБД Access 2002;
- база данных, разработанная ранее в среде Clarion 3.0;
- база данных, разработанная ранее в среде FoxPro 2.5.

Две последние БД конвертируются в Access 2002, и дальнейшая работа с ними рассматривается именно в этой единой программной среде.

Отметим, что из-за разнообразия и неформализованности информации, которая относится к предметной области, такие сведения значительно труднее обрабатывать, чем данные, связанные с большинством производственных процессов. Примеры, подтверждающие это положение, приводятся далее.

Что понимать под структуризацией информации

Как правило, банк данных аккумулирует сведения, относящиеся к определенной предметной области, то есть определенной совокупности объектов. Объектом может быть что угодно: предмет, понятие, территория, процесс, явление, фраза, связный текстовый фрагмент и т. п. Любая информация, которая накапливается в банках данных, так или иначе относится к одному из двух основных типов. По характеру объектов, образующих предметную область, эти типы условно можно назвать так:

- *фактографическая информация*, то есть данные, которые описывают конкретные факты. Такие сведения имеют количественное или логическое выражение. В настоящей книге основное внимание будет уделено работе именно с этим типом данных;
- *библиографическая информация*, то есть данные, которые очень трудно, а порой и невозможно строго классифицировать: художественная и юридическая литература, газетно-журнальные тексты и т. д.

Итак, речь идет о предварительной структуризации информации – особом этапе работы, который должен предшествовать проектированию базы данных. Сама по себе эта идея далеко не нова. Еще в начале 70-х годов усилиями в первую очередь Е. Кодда и К. Дейта была разработана теория информационных отношений и моделей данных, рассматривавшая, в частности, проблемы оптимальной структуры баз данных. Появление этих теоретических работ было обусловлено двумя причинами. Во-первых, СУБД, которые тогда использовались, были несовершенны. Во-вторых, существовали различные типы моделей данных: иерархическая, сетевая, реляционная. Разработчикам приходилось не только обоснованно выбирать определенную модель данных, но и уметь работать в рамках этой модели даже с несвойственными ей видами информационных отношений (например, в сетевой модели данных использовать иерархические структуры).

Сегодня практически единственным типом моделей данных являются *реляционные модели*. Современные СУБД имеют значительно больше возможностей для реализации различных информационных отношений между элементами данных. Видимо, поэтому иногда кажется, что проблема рационального представления информации в базе данных потеряла актуальность.

С точки зрения автора, это иллюзия. Вопрос о структуризации данных по-прежнему важен, меняется лишь технология его решения. Далее предлагается один из возможных способов структуризации данных.

Показатели

Рассмотрим утверждение, которое, согласно нашей классификации, принадлежит к классу фактографической информации. Например, «объем капитальных вложений равен 2,5 млн. руб.» или «стоимость „Мерседеса“ больше, чем стоимость „Жигулей“». Для этого класса данных под показателем понимается единица информации, которая включает ряд *реквизитов-признаков* и единственный *реквизит-основание*. Каждый реквизит-признак является мельчайшей неделимой информационной единицей и отражает какой-либо *атрибут* (свойство) объекта. Например, в энергетике такими реквизитами-признаками являются мощности, электростанции, линии электропередач, организации, расход топлива и т. д. Любой объект характеризуется перечнем свойств, которые выражаются через реквизиты.

Реквизит состоит из имени и значения. Именем реквизита будет название какой-либо качественной (наименование, местонахождение) или количественной характеристики объекта, явления, процесса (объем, размер и т. д.).

Значение реквизита представляет собой *элемент данных*, например: мощность (реквизит) – 500 МВт (его значение), электростанция (реквизит) – Красноярская ГЭС (значение), линия электропередач (реквизит) – Экибастуз-Центр (значение), расход топлива (реквизит) – 350 тонн (значение).

Совокупность реквизитов-признаков образует наименование показателя, а реквизит-основание представляет количественное или логическое значение показателя. Например, для приведенного выше показателя (мощность Красноярской ГЭС) реквизит-основание – 500 МВт. Очевидно, каждый реквизит-основание описывается одной фразой. В данном случае эта фраза выглядит так: «установленная мощность Красноярской ГЭС в 1998 году равна 500 МВт». (Это не значит, что вся база данных состоит из единственного предложения – такой случай представляется исключительным упрощением!) В следующем разделе будет показано, что реквизиты-признаки, в свою очередь, делятся на ряд категорий.

В общем случае ни один из реквизитов-признаков не может считаться обязательным. Характерной особенностью показателя является то, что он содержит определенный минимум информации, достаточный для создания документа. Ни один из перечисленных выше реквизитов, взятый в отдельности, не позволяет сформировать документ, а вот показатель может быть выдан в качестве справки при ответе на какой-либо запрос – скажем, о мощности Красноярской ГЭС. Верно и обратное – информационную совокупность любой сложности (отчет и т. д.) можно представить как определенную группу различных показателей.

Из сказанного ясно, зачем нужна предварительная структуризация информации пользователям, работающим с конкретной базой данных в определенной предметной области. Им необходима возможность формировать по единым правилам разнообразные запросы и получать на них ответы. (Примеры таких запросов и ответов будут приведены в главе 9.) Отсюда, между прочим, следует, что структуризация данных имеет свои разумные пределы. Разработчик банка данных, разбив исходную информацию на ряд категорий-реквизитов, уверен, что дальше делить данный реквизит не имеет смысла, потому что такие запросы пользователя маловероятны. Можно и остановиться. Однако, если впоследствии пользователю действительно потребуется задать специфический запрос, сделать это будет гораздо сложнее. Подобные варианты тоже будут рассмотрены ниже. Поэтому искусство разработчика состоит, в частности, в том, чтобы определить требуемую «золотую середину».

Необходимость структуризации

В качестве примера в книге рассматривается информация о фактически происшедших ЧС. Эти сведения могут поступать в виде сообщений по различным информационным каналам:

- по телефону из соответствующих региональных структур (телефонограммы). В этом случае информация вручную вводится в БД;
- по телефонному каналу связи, когда информация автоматически вводится в БД;
- по электронной почте, когда информация, при необходимости, может быть периформирована в памяти компьютера перед вводом в БД;
- по почте. Данные вводятся в БД вручную.

Информация поступает в самой различной форме, например в таком произвольном виде (реальное сообщение): «На ж/д станции Ангасолка ВосточноСибирской железной дороги (ВСЖД) в ночь с 23 на 24 марта 1999 г. допущен сход двух нефтеналивных цистерн по 60 тонн каждая, с разливом сырой нефти в одной из цистерн от 30 до 40 тонн. Произошло самовоспламенение. Основная часть нефти разлилась на северной части балластной призмы в кювете счетной стороны, примыкающей к горе, и в кармане водоотводной канавы объемом 3х4х3,5 м. Кроме того, разлитая нефть выгорела на ж/д полотне площадью 230х9 м. На другой стороне ж/д

полотна (на откосе) площадью 30x50 м происходило сжигание нефти под контролем пожарного надзора ВСЖД. Нефть застыла на снежном покрове двумя рукавами длиной по 100 метров и шириной 0,5 до 1 метра. Дополнительно выявлено еще два очага загрязнения площадью 5x2 и 5x10 м. К очистке рельефа местности от нефти привлечено 70 человек. Выдано предписание о ликвидации загрязнения с решением вопроса утилизации нефти. После проведения работ по зачистке загрязненной территории провести ее обследование комиссионно». (Имеется в виду, что обследование должно проводиться комиссией.)

Можно включать подобные сведения в БД в том виде, в каком они пришли. Такое решение вполне приемлемо, но только на начальном этапе. Рано или поздно поступившую информацию придется обрабатывать, а иметь дело с такими «сырыми» данными довольно трудно.

Конечно, можно регламентировать форму входных сообщений так, чтобы содержащиеся в них сведения были структурированы. Этот способ используется довольно широко, но он не гарантирует четкой формализации исходных данных. Дело в том, что первичное заполнение стандартных бланков производят рядовые сотрудники на местах, поэтому неизбежна значительная доля субъективизма. Это приводит к необходимости централизованной структуризации показателей при разработке и формировании банка данных.

Технология структуризации

Проведенные исследования показали, что обычно в обязательный минимум реквизитов-признаков входят следующие:

- **П** – процесс – основное наименование деятельности органа управления (операция, состояние). Это суть показателя (расход, остатки, поставка, капитальные вложения, мощность, ущерб и т. д.);

- **Ф** – формальная характеристика, то есть выраженный в наименовании способ расчета показателя (доля, темп роста, отклонение, сумма, прирост, среднее и средневзвешенное значения и т. п.), который может быть как относительным, так и абсолютным и тесно связан с процессом (иногда задан в нем неявно);

- **О** – объект, предмет операции; то, над чем она выполняется (материалы, изделия, полуфабрикаты, строительная продукция и т. д.);

- **Е** – единица измерения;

- **С** – субъект (тот, кто производит действия над объектом). Если, например, объект (**О**) – продукция, а основное наименование деятельности (**П**) – производство, то в роли субъекта (**С**) может выступать, например, предприятие, отрасль и т. д.;

- **В** – время (дата, период);

- **Ф** – функция управления (проектное, прогнозное или фактическое значение, норматив и т. п.).

Естественно, все многообразие реальных признаков не укладывается в приведенный краткий перечень. Поэтому каждый из названных реквизитов допускает практически неограниченное количество любых категорий-уточнений, которые должны удовлетворять единственному условию – представлять собой списки, состоящие из однородных терминов. Обычно уточняются следующие вопросы:

- где – в этом случае список уточнений характеризует *место действия*;

- как – список уточнений характеризует *обстоятельства действия*;

- какой – список уточнений характеризует *свойство*.

Сформированные таким образом списки при проектировании банка данных рассматриваются как *словари*. По сути, цель структуризации – создание словарей. При последующей разработке логической структуры БД они служат как бы осями координат, по которым организуется, «раскладывается» реальная информация.

Эти соображения, как уже говорилось, определяют ту границу, до которой имеет смысл проводить структуризацию. Если выясняется, что какие-то словосочетания слишком индивидуальны, уникальны и не поддаются классификации, их не следует включать в словари. В приведенном выше сообщении это формулировки типа «на северной части балластной призмы в кювете с четной стороны, примыкающей к горе, и в кармане водоотводной канавы»; «на другой стороне ж/д полотна (на откосе)». Для таких данных надо использовать специальные поля примечаний, прикрепленных к соответствующей конкретной записи.

При простой структуре исходной информации первый этап структуризации – выделение основных реквизитов-признаков – можно пропустить и сразу формировать словари. Однако учтите, что о простоте или сложности структуры исходной информации нельзя говорить вообще – это понятие имеет смысл только с одной точки зрения: легко ли будет пользователю получать ответы на запросы к БД. Поэтому прежде чем приступать к анализу первичной информации, подумайте: кто будет работать с проектируемой базой данных, какие сведения понадобятся пользователю и какими будут его запросы. В этом требовании нет ничего нового – это одно из классических положений проектирования баз данных. Но уже на начальных стадиях, при введении некоторой формализации в структуры данных, вы убедитесь, насколько важно следовать этому правилу.

Пример структуризации данных

Рассмотрим практический пример. Вы занимаетесь структуризацией информации при проектировании базы данных по контрольно-измерительным приборам, которые выпускаются различными фирмами. Это довольно простая БД, и каждая запись в ней выглядит так: «Прибор (название), с номером модели (номер), произведенный в (год) году фирмой (название), которая находится в стране (название) по адресу (приводится адрес) и имеет филиал по адресу (приводится адрес), предназначенный для (целевое назначение), имеющий характеристики (перечень технических характеристик), включенный в каталог под номером (номер в каталоге) и обслуживаемый менеджером (данные о менеджере), имеет цену (приводится цена)». Конечно, фраза громоздкая и не слишком гладкая. Поэтому ее стоит разбить на более простые фрагменты. Любой пользователь, заказчик или разработчик базы данных легко может внести в нее необходимые изменения. Ниже будет показано, как это делается.

Итак, информация о приборах включает следующие пункты:

- **О** (объект) – название прибора;
- **У** (уточнение сведений об объекте) – номер модели. Если при анализе сообщения возникает необходимость в нескольких уточнениях, то им можно присвоить номера;
 - **У** (уточнение сведений об объекте) – год выпуска прибора;
 - **У** (уточнение сведений об объекте) – номер прибора по каталогу;
 - **У** (уточнение сведений об объекте) – характеристика прибора, содержащая данные о его функциях, портативности, технических особенностях, весе, точности, способе питания, диапазоне измерений, совместимости с другими приборами;
- **С** (субъект) – название фирмы, производящей прибор;
- **У** (уточнение сведений о субъекте) – страна, в которой находится фирма;
- **У** (уточнение сведений о субъекте) – адрес фирмы;
- **У** (уточнение сведений о субъекте) – адрес филиала или дочерней фирмы, если такая есть;
- **У** (уточнение сведений о субъекте) – данные о менеджерах фирмы (фамилия, имя, отчество и адрес);
- **Р** (реквизит-основание) – цена прибора.

Предположим, пользователя в первую очередь интересует не только цена, но и вес прибора. Этот параметр можно выделить из общего массива «характеристика» и придать ему статус еще одного реквизита-основания. Тогда приведенная выше фраза-описание будет содержать две однородные фразы с параллельными реквизитами-основаниями – цена и вес.

В рассмотренном примере структура информации достаточно проста, и нужные словари могут быть сформированы практически сразу, на первом этапе проектирования. Создавая их и уточняя перечень основных реквизитов-признаков, руководствуйтесь следующим критерием: часто ли у пользователя будет необходимость запрашивать информацию по данному признаку. Если да, то имеет смысл выделить его как отдельный реквизит и сформировать соответствующий словарь. Такой признак называется *ключевым значением*, или *дескриптором*. В базе данных ему лучше выделить отдельный файл или поле в файле; этим вы существенно облегчите работу будущему пользователю. Конечно, если какой-либо признак «спрятан» в общем тексте, по нему тоже можно сделать запрос, но сформировать последний в этом случае сложнее.

В нашем примере можно сразу выделить те признаки, по которым следует ожидать частого обращения к базе данных:

- название прибора;
- название фирмы, производящей прибор;
- страна, в которой находится фирма;
- адрес фирмы;
- адрес филиала или дочерней фирмы;
- данные о менеджерах фирмы – фамилия, имя, отчество и адрес;
- номер модели;
- год выпуска прибора;
- номер прибора по каталогу;
- цена прибора;
- функциональное назначение прибора;
- вес прибора;
- категория прибора (переносной, портативный и т. п.);
- характеристика прибора.

Параметры, которые для пользователя второстепенны, остаются в общем тексте раздела.

Возьмем пример посложнее, который представлен в разделе «Необходимость структуризации». Здесь описание включает не одну, а несколько фраз, и анализ, подобный предыдущему, надо провести отдельно для каждой из них. В результате мы получим следующий набор признаков:

- **П** (показатели) – «выявлено», «выдано», «сжигание» и др.;
- **О1** (объект) – источники загрязнения (нефтеналивные цистерны);
- **О2** (объект) – загрязняющие вещества (нефть);
- **О3** (объект) – объекты загрязнения (рельеф местности);
- **О4** (объект) – документы (предписание о ликвидации последствий аварии);
- **У1** (уточнение места действия 1) – железнодорожные станции (Ангасолка);
- **У2** (уточнение места действия 2) – железные дороги (Восточно-Сибирская);
- **У3** (обстоятельство действия 1) – под контролем комиссии;
- **П** (примечания) – как уже говорилось, в этих полях должны содержаться данные – уточнения, специфичные для конкретных сообщений.

Таким образом, по мере накопления новых сообщений будут появляться и новые реквизиты, а количество параметров, указанных в скобках, тоже будет расти.

Проектирование логической структуры базы данных

Итак, мы определили состав дескрипторов, то есть ключевых полей для поиска, по которым чаще всего (по нашему прогнозу) будут формироваться запросы к базе данных. Теперь начнем разработку логической структуры БД. Под логической структурой понимается та совокупность файлов, содержащихся в них полей и связей между файлами, которую «видит» пользовательская программа, обрабатывающая базу данных.

Распределение полей по файлам

В предыдущем разделе мы постарались объяснить, почему и как необходимо выделять дескрипторные поля, по которым ожидаются запросы со стороны пользователя. Мы исходили из того, что каждому такому полю должен соответствовать словарь. Если вы в этом еще сомневаетесь, вспомните, что между элементами информации существуют различные типы отношений: «один-к-одному», «один-ко-многим», «многие-ко-многим». Очевидно, когда между какими-то элементами информации (полями) существует отношение «один-к-одному», они жестко и однозначно взаимосвязаны. В таком случае достаточно иметь один словарь на всю эту группу. Но тогда она должна находиться в одном файле, потому что иначе отношение «один-к-одному» не будет реализовано без применения каких-либо дополнительных средств. Как видите, логика довольно проста. Теперь у нас есть критерий для распределения полей по файлам: в одном файле следует размещать те поля, которые связаны между собой отношением «один-к-одному». Файлы, объединяющие такие группы полей, будут находиться друг с другом в отношении «один-ко-многим» и составят иерархическую структуру. Отметим, что файлы, находящиеся в отношениях типа «многие-ко-многим», не должны быть непосредственно взаимосвязанными. Обобщим сказанное в табл. 2.1.

В этой таблице символы **X** и **v** обозначают соответственно стороны «многие» и «один» в отношениях между реквизитами.

Таблица 2.1

	Номер	1	2	3	4	5	6	7	8	9	10	11	12
Название прибора	1	X	v	v	v	v	v	1	1	1	v	v	v
Название фирмы	2	1	X	v	1	1	1	1	1	1	v	v	v
Страна	3	v	1	X	1	1	1	v	v	v	v	v	v
Адрес фирмы	4	1	1	v	X	1	1	1	1	1	v	v	v
Адрес филиала	5	1	1	v	1	X	1	1	1	1	v	v	v
Менеджер	6	v	v	v	v	v	X	1	1	1	1	1	1
Номер модели	7	v	v	v	v	v	v	X	1	1	1	1	1
Год выпуска	8	v	v	v	v	v	v	1	X	1	v	1	1
Номер по каталогу	9	v	v	v	v	v	v	1	1	X	v	1	1
Назначение	10	v	v	v	v	v	v	v	v	1	X	v	v
Категория	11	v	v	v	v	v	v	1	1	1	v	X	v
Характеристика	12	u	u	u	u	u	u	1	1	1	u	u	X

Файлы и связи между ними

Из табл. 2.1 видно: чтобы формировать файлы, следует сгруппировать в них поля, представляющие реквизиты-признаки, находящиеся друг с другом, как сказано выше, в отношении «один-к-одному». Таким образом, будут созданы следующие файлы:

- Страны (содержит поле Название страны);
- Приборы (содержит поля Номер модели, Категория, Год выпуска, Характеристика, Номер по каталогу, Цена, Вес);
- Фирмы (содержит поля Название фирмы, Адрес фирмы, Адрес филиала);
- Менеджер (содержит поле Данные о менеджере);
- Назначение (содержит поле Назначение прибора);
- Типы приборов (содержит поле Название прибора).

Мы перечислили здесь основные – так сказать, «титульные» – поля, составляющие каркас конкретной таблицы. В нее могут также входить вспомогательные поля: **Примечания, Адрес** и др. Соединив эти файлы связями типа «один-ко-многим», мы получим логическую структуру базы данных, условный вид которой показан на рис. 2.1. О практической реализации таких связей речь пойдет в конце следующей главы.

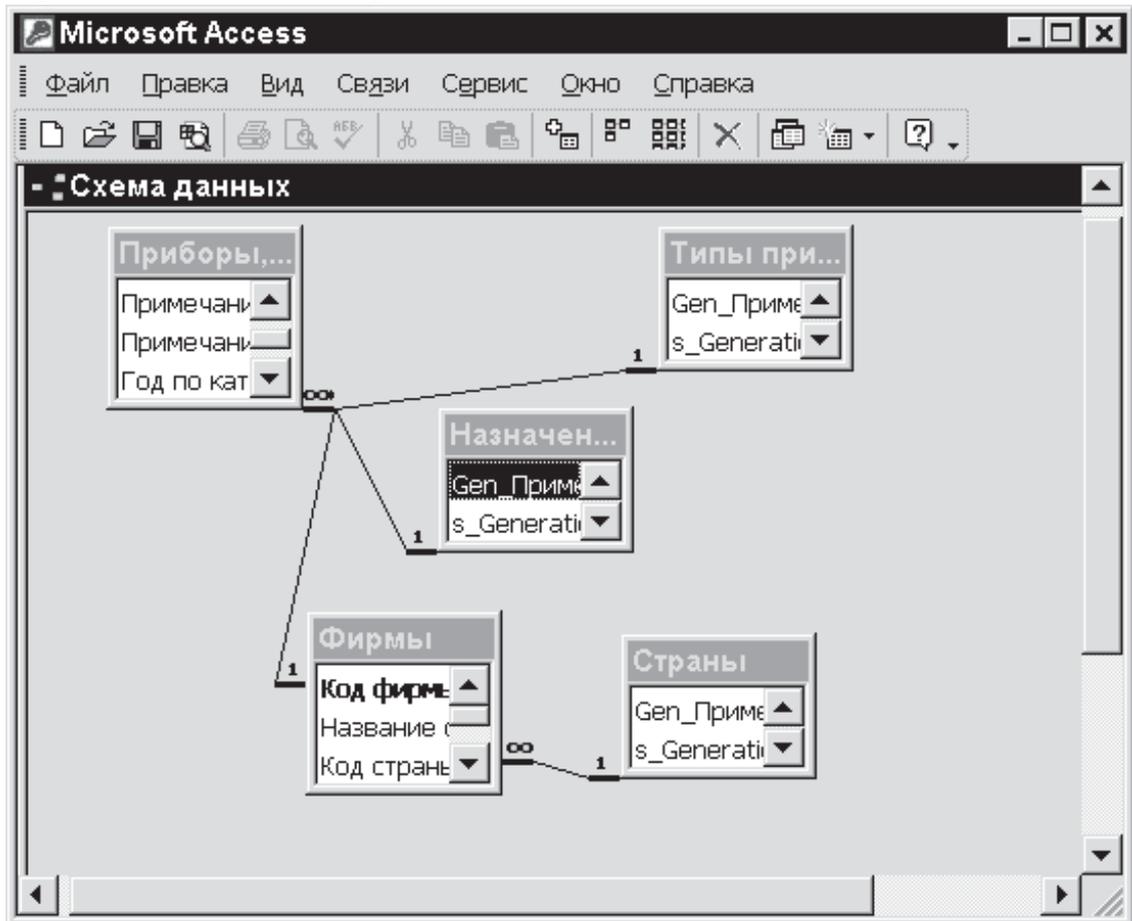


Рис. 2.1

Резюме

1. Безусловный прогресс, достигнутый в развитии программных средств СУБД и расширении их функциональных возможностей, не устранил проблему обоснованности выбора структур баз данных – от продуманности этих структур во многом зависит эффективность работы с базами данных (БД).

2. Основным элементом фактографической информации является показатель. Он, в свою очередь, состоит из множества реквизитов-признаков и единственного реквизита-основания.

3. Для того чтобы формировать по единым правилам разнообразные пользовательские запросы к БД и получать на них ответы, перед проектированием конкретных баз данных необходимо провести структуризацию информации.

4. В настоящей главе предлагается и иллюстрируется на конкретном примере технология такой структуризации и – на ее основе – последующего проектирования логической структуры БД.

Глава 3

Создание таблиц новой базы данных

Как уже было сказано в главе 2, разработка новой базы данных «Контрольно-измерительные приборы» производится в программной среде Access 2002.

Формирование БД в Access состоит из ряда последовательных этапов, описываемых ниже. Первый этап этого процесса – создание таблиц. Таблицы в Access являются теми первичными, исходными файлами, на основе которых в дальнейшем строится все здание базы данных. Access 2002, как и предыдущие версии, предоставляет пользователю несколько разных вариантов построения таблиц, а также возможность применения дополнительных аналитических табличных структур.

Порядок создания всех таблиц одинаков и не зависит от их названия и конкретного содержания. Мы рассмотрим этот процесс на примере таблицы **Страны**.

Варианты создания таблиц

Формирование таблицы начинается с того, что вы открываете окно базы данных и в нем выбираете пункт **Таблицы** в разделе **Объекты** – рис. 3.1.

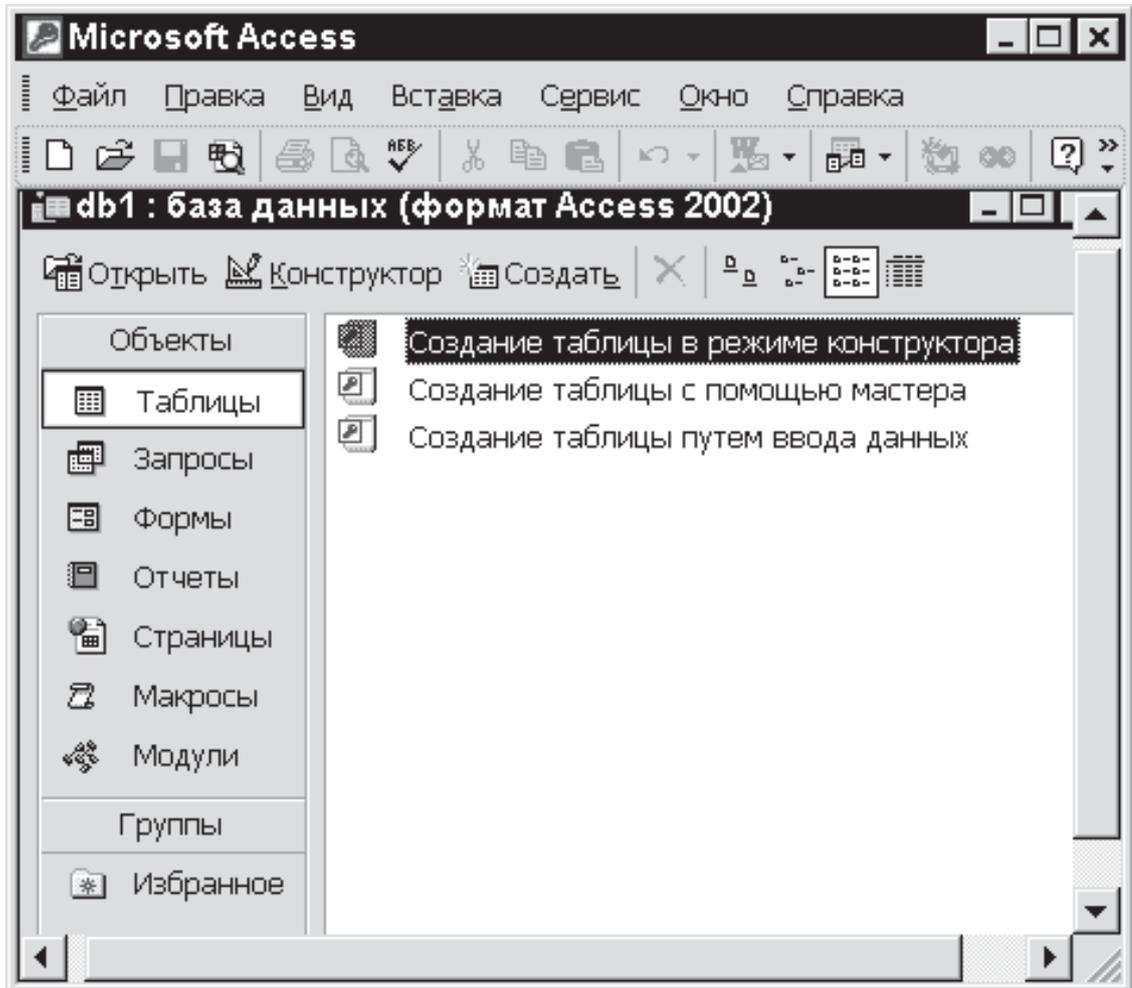


Рис. 3.1

Дальше следует щелкнуть по кнопке



на панели окна БД. В диалоговом окне **Новая таблица**, показанном на рис. 3.2, представлены все возможные параметры создания таблицы:

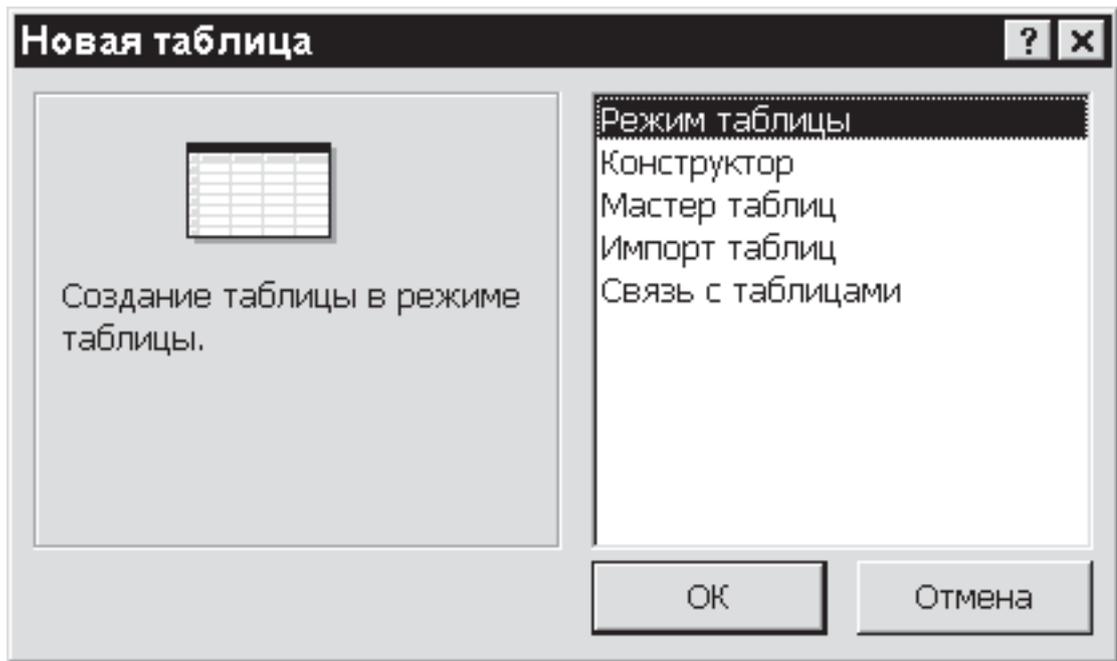


Рис. 3.2

- Режим таблицы;
- Конструктор;
- Мастер таблиц;
- Импорт таблиц;
- Связь с таблицами.

Последние два варианта создания таблиц – импорт таблиц и связь с таблицами – рассматриваются в том разделе главы 7, который посвящен объединению разнородных баз данных.

Формирование таблицы в режиме ввода

Войти в этот режим можно двумя способами: либо выбрав пункт **Режим таблицы** в окне **Новая таблица** (см. рис. 3.2) и щелкнув по кнопке **ОК**, либо выбрав опцию **Создание таблицы путем ввода данных** в окне базы данных (см. рис. 3.1). В результате на экране появится таблица, готовая к вводу информации (рис. 3.3).

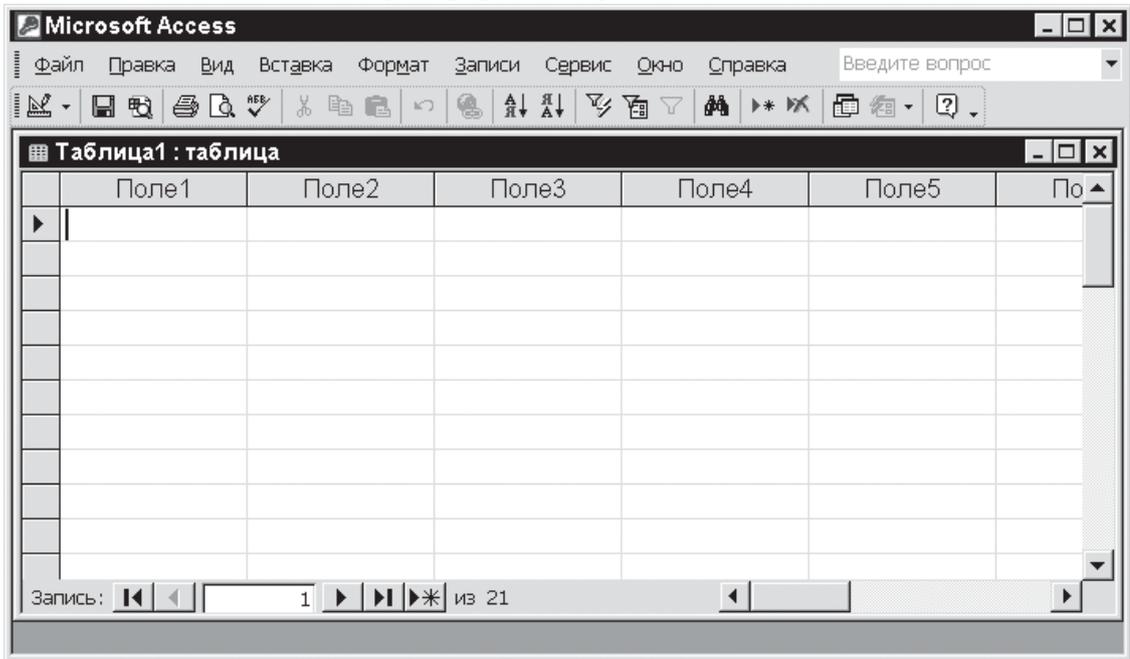


Рис. 3.3

Ввод данных

Чтобы осуществить ввод данных, сначала надо заменить имеющиеся заголовки столбцов, а затем уже ввести сведения в поля таблицы. Рассмотрим эту операцию на примере создания таблицы **Страны**.

Заменим имена полей **Поле1** и **Поле2** на **Код** и **Страна**. Для этого дважды щелкните мышью в ячейке с именем соответствующего поля, а затем введите нужные значения. Записав первые данные (см. рис. 3.4), попробуйте выйти из созданной таблицы (кнопка в правом верхнем углу). Сначала Access 2002 спросит вас, надо ли сохранять произведенные в таблице изменения (если вы не хотите этого делать, она вообще сотрется из памяти). Затем вам будет предложено назвать таблицу (или согласиться с предлагаемым именем, которое присваивается автоматически). Все таблицы программа называет именем **Таблица** с добавлением текущего номера.

Рис. 3.5

Если вы отказываетесь установить ключевые поля и отвечаете **Нет**, программа запомнит таблицу в том виде, в каком она показана на рис. 3.4. Однако вопрос о ключевых полях все равно возникнет на следующем этапе работы – когда вы будете формировать связи между файлами, и уж тогда без ключевых полей система не справится с этой задачей. Если вы поддались на уговоры и сразу согласились создать первичный код, выбрав ответ **Да**, то программа сама сделает это. В результате таблица будет выглядеть так, как показано на рис. 3.6.

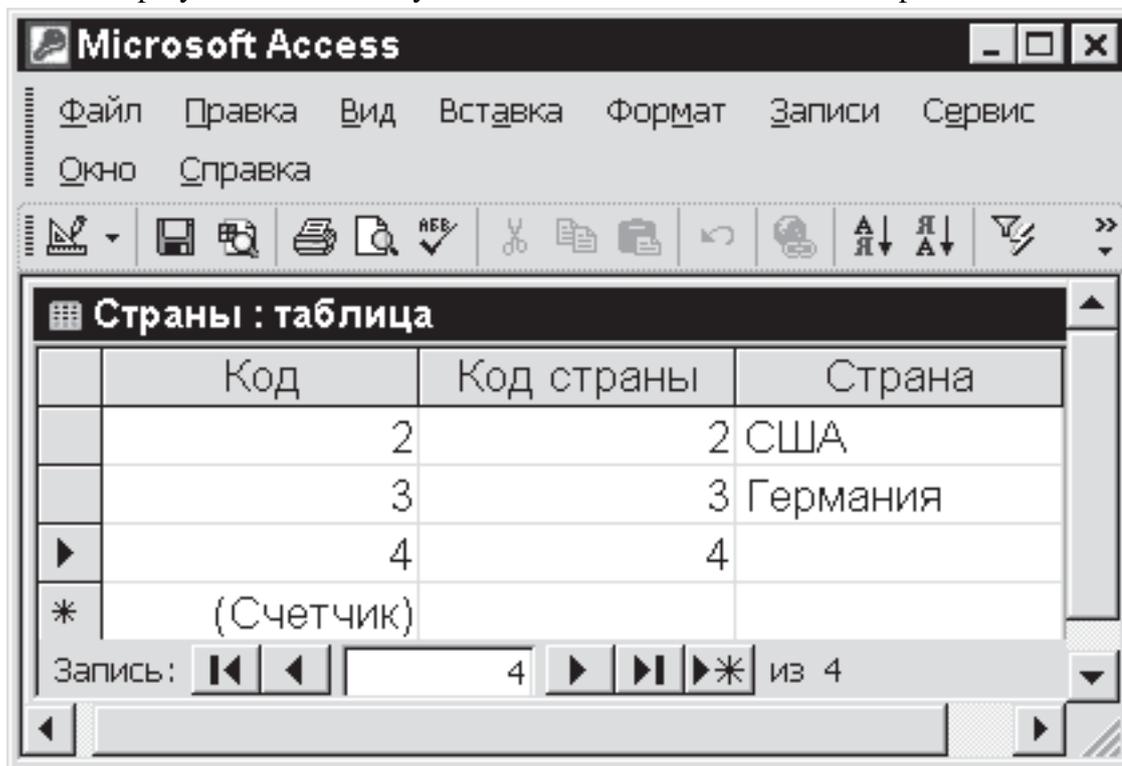


Рис. 3.6

Если вы выберете пункт **Отмена**, таблица вновь примет тот вид, который показан на рис. 3.4. Однако это промежуточное состояние, из которого все равно надо как-то выходить. Внимательно посмотрите на первичные коды, созданные системой в поле **Код**. Здесь они ничем не отличаются от кодов, созданных пользователем в поле **Код страны**. Но в общем случае коды, введенные в это поле, совсем не обязаны быть такими же упорядоченными, как коды поля **Код**, – таблица, показанная на рис. 3.6, представляет собой словарь, и коды могут периодически изменяться. Поэтому для надежного контроля за файлами в Access предусмотрен механизм системных первичных кодов. Иногда (как, например, сейчас) они вводятся только по желанию пользователя. В других случаях при отсутствии этих кодов ряд функций Access 2002 выполняться не будет.

Поэтому, если у пользователя нет каких-то серьезных причин отказываться от введения первичных кодов, желательно их все-таки создать.

Отметим, что в словарном файле совсем не обязательно ограничиваться тем минимумом полей, которые показаны в нашем примере. Это было сделано исключительно в методических целях. На практике количество полей в файле вообще не ограничивается, следует лишь соблюдать единственное требование: поля должны находиться в отношениях «один-к-одному».

Создание таблицы в режиме конструктора

Описанная в предыдущем разделе таблица создавалась, можно сказать, стихийно. Теперь мы будем придерживаться строгой последовательности действий, заблаговременно обосновывая все дальнейшие операции, задавая необходимые данные и их форматы.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.