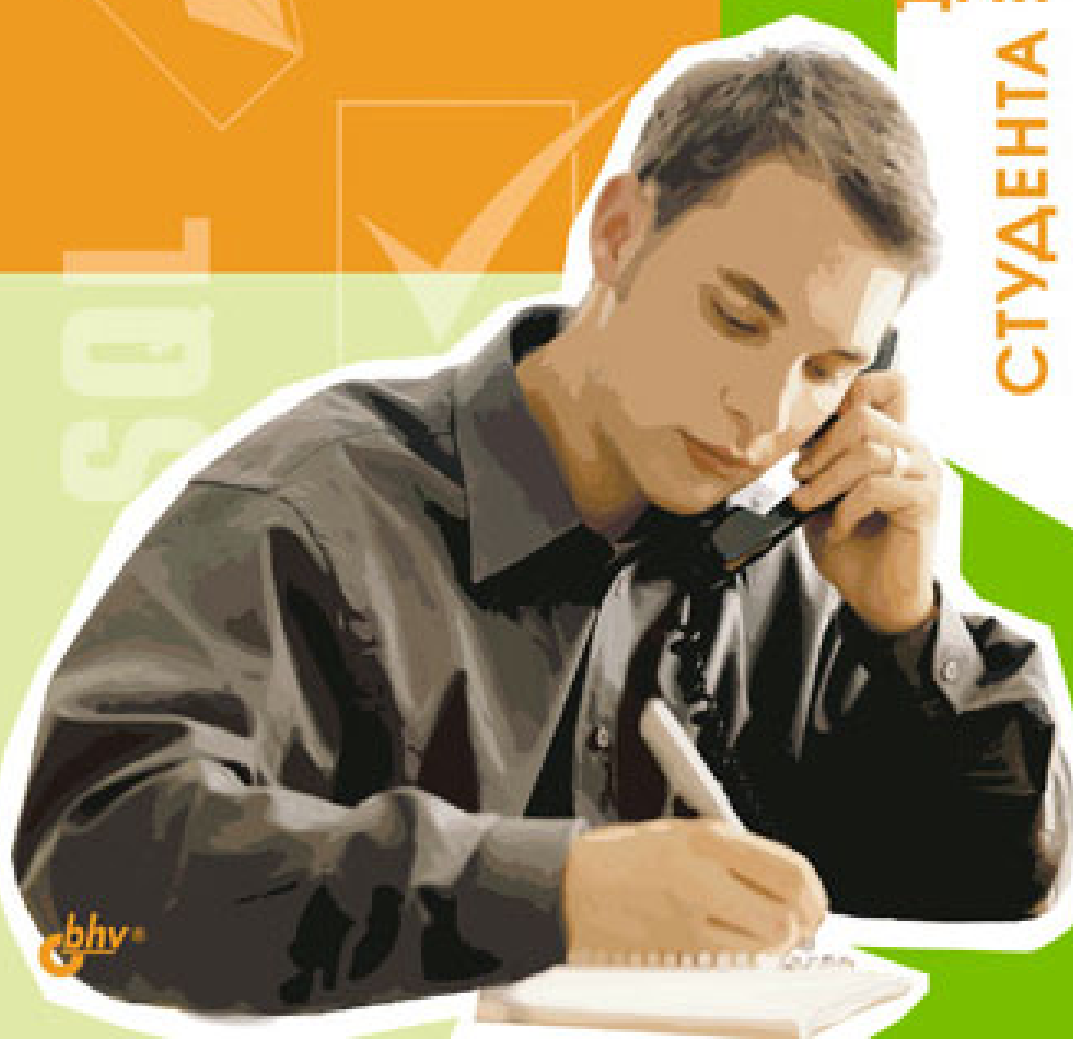


В. В. Дунаев

БАЗЫ ДАННЫХ ЯЗЫК SQL

ДЛЯ
СТУДЕНТА



Вадим Дунаев

БАЗЫ ДАННЫХ ЯЗЫК SQL

ДЛЯ СТУДЕНТА

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.068+800.92SQL
ББК 32.973.26-018.1
Д83

Дунаев В. В.

Д83 Базы данных. Язык SQL. — СПб.: БХВ-Петербург, 2006. — 288 с.: ил.

ISBN 978-5-94157-823-8

Рассмотрен язык структурированных запросов для взаимодействия с базами данных — SQL — начиная с доступного изложения теории отношений (реляционной теории) и заканчивая вопросами администрирования СУБД с помощью запросов. На практических примерах подробно описаны основные конструкции языка, а также различные типы запросов: простые, сложные, рекурсивные. Показано, как осуществлять вычисления в запросах с помощью агрегатных функций и условных выражений. Рассмотрены операции над наборами записей, соединение таблиц, модификация данных, транзакции, курсоры, хранимые процедуры и функции. Уделено внимание администрированию СУБД с помощью запросов. Материал сопровождается задачами для самостоятельного решения.

Для студентов и программистов

УДК 681.3.068+800.92SQL
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Нина Седых</i>
Компьютерная верстка	<i>Натали Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн серии	<i>Игоря Цырульников</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.02.06.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 18.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-94157-823-8

© Дунаев В. В., 2006
© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

От автора	1
Благодарности.....	1
Введение.....	3
Как устроена эта книга.....	4
Как выполнять примеры SQL-выражений.....	6
Работа с Microsoft SQL Server 2000.....	7
Работа с Microsoft Access 2003.....	11
Глава 1. Основы реляционных баз данных	17
1.1. Множества.....	17
1.2. Отношения.....	29
1.2.1. Общие сведения.....	29
1.2.2. Способы представления отношений	32
1.2.3. Операции над отношениями	35
1.3. Декомпозиция отношений.....	42
1.3.1. Корректная декомпозиция	43
1.3.2. Пример некорректной декомпозиции.....	44
1.3.3. Зависимости между атрибутами.....	45
1.3.4. Правила вывода зависимостей	55
1.3.5. Ключи.....	57
1.4. Ограничения целостности отношений.....	60
1.4.1. Семантическая целостность	60
1.4.2. Доменная целостность	61
1.4.3. Ссылочная целостность	61
1.5. Нормализация таблиц.....	63
1.5.1. Первая нормальная форма.....	65
1.5.2. Вторая нормальная форма	66

1.5.3. Третья нормальная форма	67
1.5.4. Доменно-ключевая нормальная форма.....	67
1.5.5. Денормализация.....	68
Глава 2. Основы SQL	69
2.1. Что такое SQL.....	69
2.2. Типы данных.....	73
2.2.1. Строки.....	76
2.2.2. Числа.....	77
2.2.3. Логические данные.....	80
2.2.4. Дата и время	80
2.2.5. Интервалы	83
2.2.6. Специальные типы данных	84
2.2.7. Пользовательские типы данных.....	86
2.2.8. Неопределенные значения	90
2.2.9. Преобразование типов	91
Глава 3. Простые выборки данных	93
3.1. Основное SQL-выражение для выборки данных	94
3.2. Уточнения запроса.....	98
3.2.1. Оператор <i>WHERE</i>	102
3.2.2. Оператор <i>GROUP BY</i>	109
3.2.3. Оператор <i>HAVING</i>	111
3.2.4. Оператор <i>ORDER BY</i>	112
3.2.5. Логические операторы	113
3.3. Задачи	114
Задача 3.1	115
Задача 3.2.....	115
Задача 3.3.....	115
Глава 4. Вычисления.....	117
4.1. Итоговые функции.....	117
4.2. Функции обработки значений.....	121
4.2.1. Строковые функции.....	121
4.2.2. Числовые функции	123
4.2.3. Функции даты-времени	125
4.3. Вычисляемые выражения.....	126

4.4. Условные выражения с оператором <i>CASE</i>	129
4.4.1. Оператор <i>CASE</i> со значениями	129
4.4.2. Оператор <i>CASE</i> с условиями поиска	130
4.4.3. Функции <i>NULLIF</i> и <i>COALESCE</i>	132

Глава 5. Сложные запросы..... 135

5.1. Подзапросы	136
5.1.1. Простые подзапросы	136
5.1.2. Связанные подзапросы	142
5.2. Теоретико-множественные операции	147
5.2.1. Декартово произведение наборов записей	147
5.2.2. Объединение наборов записей (<i>UNION</i>)	149
5.2.3. Пересечение наборов записей (<i>INTERSECT</i>)	152
5.2.4. Вычитание наборов записей (<i>EXCEPT</i>)	153
5.3. Операции соединения	154
5.3.1. Естественное соединение (<i>NATURAL JOIN</i>)	155
5.3.2. Условное соединение (<i>JOIN ... ON</i>)	157
5.3.3. Соединение по именам столбцов (<i>JOIN ... USING</i>)	157
5.3.4. Внешние соединения	159
5.3.5. Рекурсивные запросы.....	164
5.4. Задачи	167
Задача 5.1	167
Задача 5.2	170
Задача 5.3	170
Задача 5.4	171
Задача 5.5	171

Глава 6. Добавление, удаление и изменение данных в таблицах..... 173

6.1. Добавление новых записей	173
6.2. Удаление записей	176
6.3. Изменение данных.....	178
6.4. Проверка ссылочной целостности	182

Глава 7. Создание и модификация таблиц..... 185

7.1. Создание таблиц.....	185
7.1.1. Ограничения для столбцов.....	187
7.1.2. Ограничения для таблиц.....	190

7.1.3. Внешние ключи	192
7.2. Удаление таблиц	195
7.3. Модификация таблиц	196
7.4. Представления	200
7.4.1. Что такое представление	200
7.4.2. Создание представлений	203
7.4.3. Изменение данных в представлениях	206
7.5. Задачи	207
Задача 7.1	207
Задача 7.2	208
Задача 7.3	208
Задача 7.4	208
Глава 8. Транзакции	209
8.1. Как устроена транзакция	210
8.2. Определение параметров транзакции	212
8.3. Уровни изоляции транзакций	213
8.3.1. Неподтвержденное чтение	213
8.3.2. Подтвержденное чтение	214
8.3.3. Повторяющееся чтение	214
8.3.4. Последовательное выполнение	215
8.4. Субтранзакции	215
8.5. Ограничения в транзакциях	216
Глава 9. Курсоры и применение SQL в приложениях	221
9.1. Объявление курсора	224
9.1.1. Чувствительность	224
9.1.2. Перемещаемость	225
9.1.3. Выражение запроса	226
9.1.4. Сортировка	226
9.1.5. Разрешение обновления	227
9.2. Открытие и закрытие курсора	228
9.3. Работа с отдельными записями	229
9.4. SQL в приложениях	232
Глава 10. Постоянно хранимые модули	237
10.1. Составные команды	238
10.1.1. Атомарность	239
10.1.2. Переменные	240

10.1.3. Обработка состояния.....	241
10.2. Операторы условного перехода	244
10.2.1. Оператор <i>IF</i>	244
10.2.2. Оператор <i>CASE ... END CASE</i>	245
10.3. Операторы цикла.....	247
10.3.1. Оператор <i>LOOP...END LOOP</i>	247
10.3.2. Оператор <i>WHILE ... DO ... END WHILE</i>	248
10.3.3. Оператор <i>REPEAT ... UNTIL ... END REPEATE</i>	249
10.3.4. Оператор <i>FOR ... DO ... END FOR</i>	249
10.3.5. Оператор <i>ITERATE</i>	251
10.4. Хранимые процедуры и функции	252
10.5. Хранимые модули.....	255

Глава 11. Управление правами доступа..... 257

11.1. Пользователи	257
11.1.1. Администратор базы данных.....	258
11.1.2. Владелец объектов базы данных	258
11.1.3. Другие пользователи.....	259
11.1.4. Создание пользователей.....	259
11.2. Предоставление привилегий	260
11.2.1. Роли и группы	262
11.2.2. Право просмотра данных.....	263
11.2.3. Право изменять данные	263
11.2.4. Право удалять записи	264
11.2.5. Право на использование ссылок.....	264
11.2.6. Право на домены	265
11.2.7. Право предоставлять права.....	266
11.3. Отмена привилегий.....	267

Приложение. Зарезервированные слова SQL..... 271

Предметный указатель 275

Введение

SQL (Structured Query Language) — это структурированный язык запросов к реляционным базам данных. На этом языке можно формулировать выражения (запросы), которые извлекают требуемые данные, модифицируют их, создают таблицы и изменяют их структуры, определяют права доступа к данным и многое другое.

Запросы выполняются системой управления базой данных (СУБД). Если вы не являетесь специалистом по разработке и администрированию баз данных, то вполне можете быть их пользователем, который просматривает или/и изменяет данные в уже имеющихся таблицах. Во многих случаях эти и другие операции с базой данных выполняются с помощью специальных приложений, предоставляющих пользователю удобный интерфейс. Обычно приложения пишутся на специальных языках программирования (C, Pascal, Visual Basic и т. п.) и чаще всего создаются с помощью интегрированных сред разработки, например, Delphi, C++ Builder и др. Однако доступ к базе данных можно получить и без них — с помощью только SQL. Замечу также, что и специализированные приложения обычно используют SQL-фрагменты кода при обращениях к базе данных.

Таким образом, SQL — широко распространенный стандартный язык работы с реляционными базами данных. Синтаксис этого языка достаточно прост, чтобы его могли использовать рядовые пользователи, а не только программисты. В настоящее время обычный пользователь компьютера должен владеть, по крайней мере, текстовым редактором (например, Microsoft Word) и электронными таблицами (например, Microsoft Excel). Неплохо, если

он также умеет пользоваться базами данных. Различных СУБД существует много, а универсальное средство работы с базами данных одно — SQL. Знание SQL, хотя бы его основ, и умение его применять для поиска и анализа данных является фундаментальной частью компьютерной грамотности даже рядовых пользователей.

Как устроена эта книга

Данная книга посвящена языку манипулирования реляционными базами данных, а не их проектированию и сопровождению. Однако успех изучения и применения SQL существенно зависит от понимания того, как устроена реляционная база данных. Теоретическим источником реляционных баз данных является теория отношений (реляционная теория). Основам этой теории посвящена *глава 1*. Несмотря на обилие символики, изложенный в ней материал достаточно прост и вполне доступен широкому кругу читателей. Однако те, кого интересуют непосредственно SQL и возможность быстрее получить практический результат, могут пропустить эту главу при первом чтении. Изложение последующих глав построено в основном на рассмотрении примеров.

Таблицы, из которых состоит любая реляционная база данных, представляют собой некоторые отношения, а отношения являются не чем иным, как множествами записей (строк). Все запросы к базе данных, направленные на извлечение из нее нужных сведений, интерпретируются как инструкции по выполнению тех или иных операций, являющихся в конечном счете операциями алгебры множеств. Более или менее серьезные базы данных состоят из нескольких таблиц, между которыми могут существовать связи. Рассмотрение в реляционной теории декомпозиции одной таблицы на несколько других позволит вам понять противоположный процесс — проектирование базы данных как композиции нескольких таблиц.

Главы 2—4 содержат минимальный набор сведений, позволяющий в большинстве случаев извлекать из базы данных необходимую информацию.

В *главе 2* кратко рассматривается история и составные части SQL, а также типы данных. При первом прочтении желательно

получить хотя бы поверхностное представление о типах данных. Дело в том, что наборы типов данных SQL и типов столбцов таблиц, поддерживаемых той или иной СУБД, как правило, не совпадают. Однако между большинством из них есть соответствие. Кроме того, можно использовать функцию приведения к заданному типу.

В *главе 3* описываются запросы на выборку данных, которые наиболее часто используются на практике. Это так называемые простые запросы, не содержащие вложенных запросов. Здесь же рассматриваются условия, которые могут применяться не только в запросах на выборку, но и при модификации данных.

Глава 4 посвящена вычислениям в запросах с помощью итоговых (агрегатных) функций и условных выражений.

В *главе 5* рассматриваются так называемые сложные запросы, а также теоретико-множественные операции над наборами записей, соединения таблиц и рекурсивные запросы. Нередко для получения данных приходится вначале выполнить некий вспомогательный запрос, чтобы затем его результат использовать при формулировке условия выборки данных в основном запросе. Вспомогательный запрос включается в выражение основного запроса, который называют сложным (содержащим подзапрос). Теоретико-множественные операции позволяют из нескольких наборов записей получить другой набор записей — объединение, пересечение или разность исходных наборов. Операция соединения таблиц дает в результате таблицу, записи в которой получают путем некоторой комбинации записей соединяемых таблиц.

Модификация (изменение, добавление и удаление) данных описывается в *главе 6*.

Рано или поздно возникает задача создания новых или изменения структуры уже существующих таблиц. Средства SQL, позволяющие это сделать, рассмотрены в *главе 7*. Здесь же описываются представления — виртуальные таблицы, доступные многим пользователям.

При выполнении многоэтапных операций с базами данных, особенно в многопользовательском режиме, последовательности SQL-выражений объединяются в транзакции. Если какое-либо SQL-выражение в транзакции по какой-то причине не выполнилось, отменяется действие всех SQL-выражений в этой транзакции,

а база данных возвращается в исходное состояние, в котором она находилась до начала транзакции. Иначе говоря, выполняется принцип "все или ничего". Транзакции описываются в *главе 8*.

В *главе 9* рассмотрены так называемые курсоры. Курсор позволяет сделать обработку одной или нескольких записей таблицы с помощью SQL-выражений и, таким образом, упрощает совместное использование SQL с другими языками программирования, на которых пишутся приложения баз данных. Выборка записей производится с помощью SQL-запроса, а проверка их содержимого — с помощью кода на процедурном языке.

Глава 10 посвящена постоянно хранимым модулям. На языке SQL можно написать процедуры и функции, которые будут содержать объявления переменных и составные команды SQL. Коды этих процедур и функций могут содержать управляющие структуры, а также обычные SQL-выражения. Функции и процедуры задаются в рамках так называемого модуля, который представляет собой контейнер для их размещения. Модуль создается специальной командой SQL и сохраняется в метаданных базы данных, т. е. становится ее компонентом, подобно таблицам, индексам и т. д. Таким образом, однажды создав модуль с процедурами и функциями, вы можете затем в частных SQL-запросах использовать их вызовы.

В *главе 11* рассматриваются некоторые возможности SQL по администрированию базы данных, а именно, по предоставлению прав доступа к ее объектам. Разграничение прав доступа является важным средством защиты базы данных от неправильного использования содержащейся в ней информации различными категориями пользователей.

В конце некоторых глав приводятся задачи для самоконтроля усвоения прочитанного материала. Список зарезервированных слов SQL приведен в приложении.

Как выполнять примеры SQL-выражений

Чтобы выполнить примеры и решить задачи, приведенные в книге, необходимо установить на компьютере какую-нибудь СУБД, которых существует множество. Наиболее простым вариантом является Microsoft Access. Однако не все возможности SQL-92

поддерживаются этой СУБД. Я рекомендую установить какой-нибудь SQL-сервер, например, Microsoft SQL Server 2000 или PostgreSQL. Далее мы кратко рассмотрим создание базы данных и SQL-выражений с помощью SQL Server 2000 и Access 2003.

Работа с Microsoft SQL Server 2000

Установка Microsoft SQL Server 2000

Процедура инсталляции SQL Server 2000 с установочного CD-диска довольно проста. Если инсталлятор не запустился автоматически, то следует запустить программу \x86\setup\setupsql.exe, расположенную на установочном CD. Далее мастер предоставит вам возможность указать параметры установки. Один из возможных вариантов установки, пригодный для большинства новичков, работающих на локальном компьютере, выглядит следующим образом:

1. **Local Computer** (Локальный компьютер).
2. **Server and Client Tools** (Инструменты сервера и клиента). При этом будет инсталлирован сам сервер и средства администрирования.
3. Настройка учетных записей служб:
 - **Use the same account for each service. Auto start SQL Server Service** (Использовать одну учетную запись для всех служб с их автоматическим запуском при загрузке операционной системы);
 - **Use the Local System account** (Использовать локальную учетную запись);
 - **Auto Start Service** (Автоматический запуск служб при загрузке операционной системы).
4. **Typical** (Установка всех компонентов).

После инсталляции сервера баз данных в меню Windows **Пуск | Программы | Microsoft SQL Server** запустите утилиту Enterprise Manager, которая предоставляет пользовательский интерфейс для настройки, управления и доступа к данным на сервере (рис. В1). В левой части окна утилиты отображается древовидный список, в котором щелчком левой кнопкой мыши следует

раскрыть узел **Microsoft SQL Servers/SQL Server Group**. Если установка прошла успешно, то в этом узле должен находиться узел, имя которого состоит из имени вашей учетной записи, за которым следует "(Windows NT)". На рис. В1 это **DVV (Windows NT)**. В правой части окна отображаются элементы того узла, который выделен в левой части окна (в дереве).

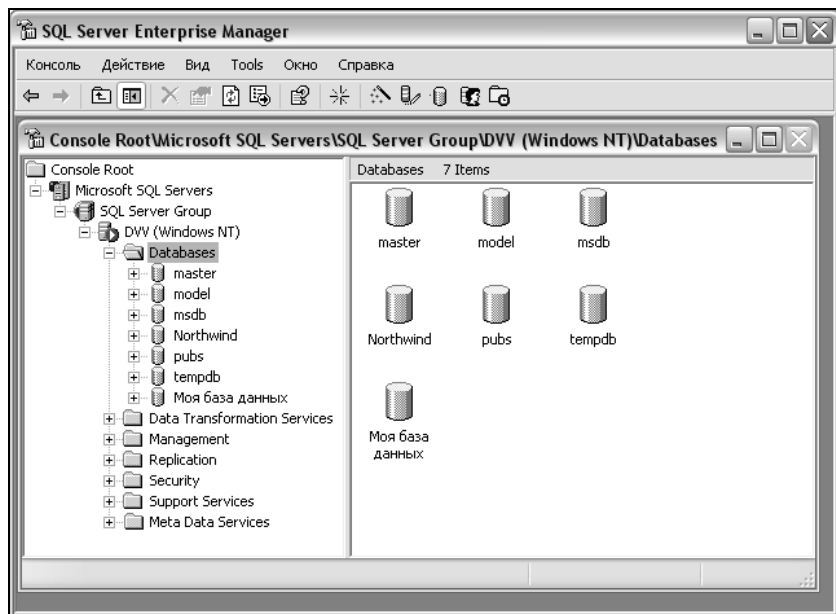


Рис. В1. Окно утилиты Enterprise Manager

Создание базы данных в Enterprise Manager

Чтобы выполнить учебные примеры SQL, необходимо создать учебную базу данных, на которой вы будете экспериментировать. Для этого в древовидном списке окна утилиты Enterprise Manager раскройте узел, соответствующий вашей учетной записи (на рис. В1 **DVV (Windows NT)**). Среди содержащихся в нем подузлов имеется **Databases** (Базы данных). Щелкните на нем правой кнопкой мыши и в раскрывшемся контекстном меню выберите **New Database** (Новая база данных). В результате откроется диалоговое окно **Database Properties** (Свойства базы данных).

В поле **Name** (Имя) введите имя базы данных (например, "Моя база данных") и щелкните на кнопке **ОК**. В результате в узле **Databases** появится узел с именем, совпадающим с именем базы данных.

Итак, создана пустая база данных, которая пока не содержит ни одной таблицы с пользовательскими данными. Теперь создайте какую-нибудь таблицу, например, **Клиенты** с символьными столбцами **Имя**, **Адрес**, **Телефон**. Для этого щелкните правой кнопкой мыши на имени вашей базы данных в древовидном списке и в контекстном меню выберите **Создать | Table**. В результате откроется окно **New Table** (Новая таблица), показанное на рис. В2. В этом окне следует ввести имена столбцов (**Column Name**), тип данных (**Data Type**) и длину данных (**Length**). В рассматриваемом примере все столбцы являются символьными (строковыми, текстовыми), т. е. имеют тип `char`.

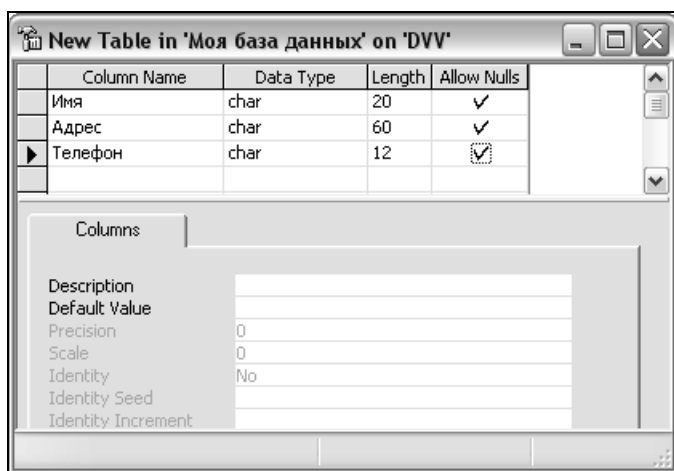


Рис. В2. Окно **New Table** для определения столбцов таблицы

После завершения определения столбцов (структуры) создаваемой таблицы закройте окно, после чего появится запрос о сохранении введенных данных. Подтвердите необходимость сохранения и в открывшемся диалоговом окне введите имя созданной таблицы (например, **Клиенты**), а затем щелкните на кнопке **ОК**.

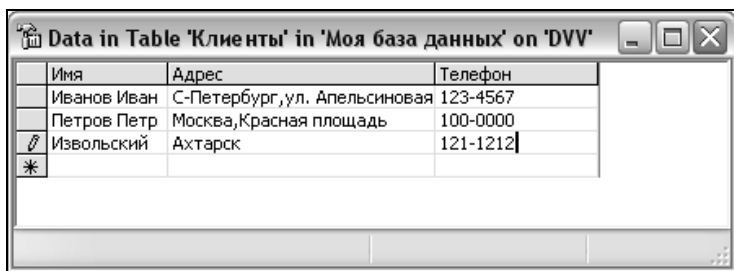
Если в древовидном списке раскрыть узел с именем вашей базы данных и щелкнуть на узле **Tables** (Таблицы), то в правой части

окна Enterprise Manager отобразятся все таблицы, входящие в эту базу данных. Среди множества служебных таблиц вы найдете и свою только что созданную таблицу.

Вы можете изменить структуру созданной таблицы или/и изменить содержащиеся в ней данные.

Для изменения структуры таблицы щелкните правой кнопкой мыши на ее имени и в контекстном меню выберите **Design Table** (Разработка таблицы). В результате откроется окно, аналогичное показанному на рис. В2. Теперь можно добавить новые или удалить имеющиеся столбцы, изменить их имена и/или другие параметры.

Для ввода, редактирования и удаления данных в таблице щелкните правой кнопкой мыши на ее имени и в контекстном меню выберите **Open Table | Return all rows** (Открыть таблицу | Вернуть все записи). В результате откроется окно, показанное на рис. В3. В этом окне можно вводить и изменять значения столбцов, добавлять и удалять строки (записи). Поскольку новая таблица пуста, то начать следует с добавления записей.



Имя	Адрес	Телефон
Иванов Иван	С-Петербург, ул. Апельсиновая	123-4567
Петров Петр	Москва, Красная площадь	100-0000
Извольский	Ахтарск	121-1212

Рис. В3. Окно для модификации данных таблицы

Аналогичным образом можно создать и другие таблицы базы данных.

Создание и выполнение SQL-выражений в SQL Query Analyser

Для создания, редактирования и выполнения выражений на языке SQL рекомендуется использовать утилиту SQL Query Analyser. Эту утилиту можно вызвать из меню **Tools** (Инструменты),

расположенного в верхней части окна Enterprise Manager. В результате откроется окно, показанное на рис. В4. Здесь можно ввести, как в обычном текстовом редакторе, SQL-выражение.

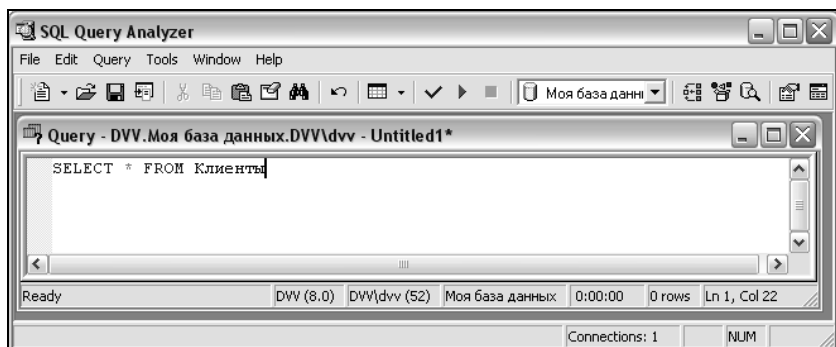


Рис. В4. Окно утилиты SQL Query Analyzer

Прежде чем выполнить введенное SQL-выражение, рекомендуется его проверить, щелкнув на кнопке **Parse Query** (Анализировать запрос) с изображением галочки или нажав клавиши <Ctrl>+<F5>. Для выполнения выражения достаточно щелкнуть на кнопке с изображением стрелки вправо или нажать клавишу <F5>.

При желании созданное SQL-выражение можно сохранить в текстовом файле с расширением sql. Сохраненные в файлах SQL-выражения можно снова открыть в окне SQL Query Analyzer. Соответствующие команды выбираются из меню **File**.

Работа с Microsoft Access 2003

Создание базы данных

При запуске программы Access открывается главное окно, в правой части которого находится панель **Приступая к работе**. На этой панели в разделе **Открыть** выберите команду **Создать файл**. В результате отобразится панель **Создание файла**. В разделе **Создание** на этой панели выберите команду **Новая база данных**. В открывшемся диалоговом окне **Файл новой базы данных** введите имя файла создаваемой базы данных (например, "Моя база

данных") и сохраните его в какой-нибудь папке. Имя файла базы данных в Access имеет расширение mdb.

Итак, пустая база данных создана. В главном окне Access откроется окно **Моя база данных: база данных**. В этом окне выберите **Создание таблицы в режиме конструктора**. Откроется окно **Таблица1: таблица**, показанное на рис. В5. В этом окне определяется структура (свойства столбцов) таблицы — имена, типы, размеры и другие параметры столбцов.

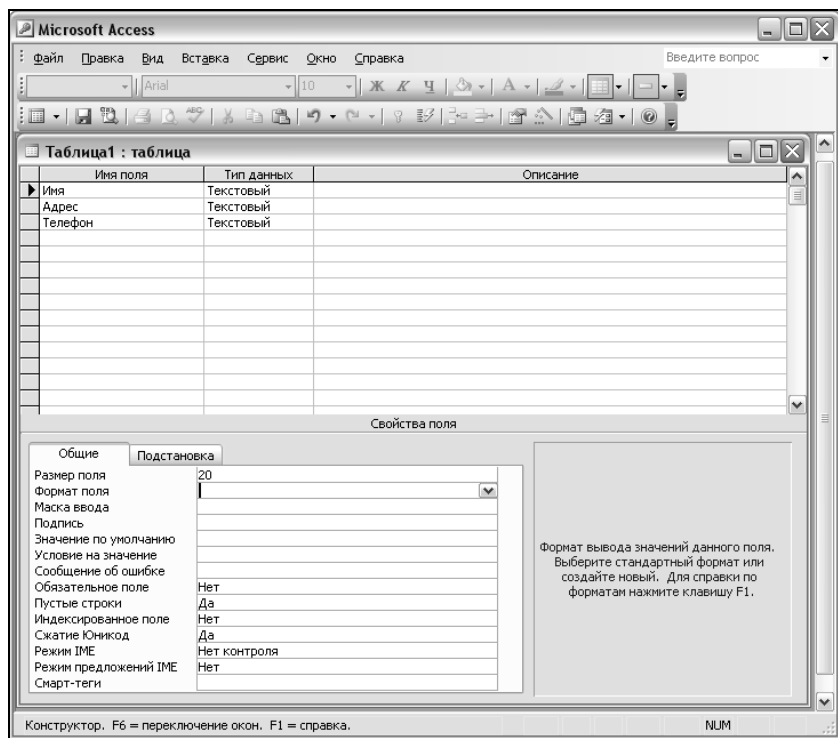


Рис. В5. Окно **Таблица1: таблица** для определения столбцов таблицы

По завершении определения столбцов (структуры) создаваемой таблицы закройте это окно, после чего появится запрос о сохранении введенных данных. Подтвердите сохранение и в открывшемся диалоговом окне введите имя созданной таблицы (например, **Клиенты**) и щелкните на кнопке **ОК**.

В окне **Моя база данных: база данных** появится пиктограмма с именем созданной таблицы. Вы можете изменить структуру созданной таблицы или/и изменить содержащиеся в ней данные.

Для изменения структуры таблицы щелкните правой кнопкой мыши на ее имени и в контекстном меню выберите **Конструктор**. В результате откроется окно, аналогичное показанному на рис. В5. Теперь можно добавлять новые или удалять имеющиеся столбцы, изменять их имена и/или другие параметры.

Двойной щелчок на пиктограмме с именем таблицы откроет ее в режиме редактирования данных (рис. В6). В этом окне можно вводить и изменять значения столбцов, добавлять и удалять строки (записи). Поскольку новая таблица пуста, то следует начать с добавления записей.



Рис. В6. Окно для модификации данных таблицы

Аналогичным образом можно создать и другие таблицы базы данных.

Создание и выполнение SQL-выражений

В окне **Моя база данных: база данных** (см. рис. В6) в списке объектов выберите **Запросы**, а в области задач, расположенной в правой части этого окна, выполните двойной щелчок на опции **Создание запроса в режиме конструктора**. В результате откроются два окна: **Запрос1: запрос на выборку** и **Добавление таблицы**. Окно **Добавление таблицы** закройте, после чего на экране останутся окна, показанные на рис. В7.

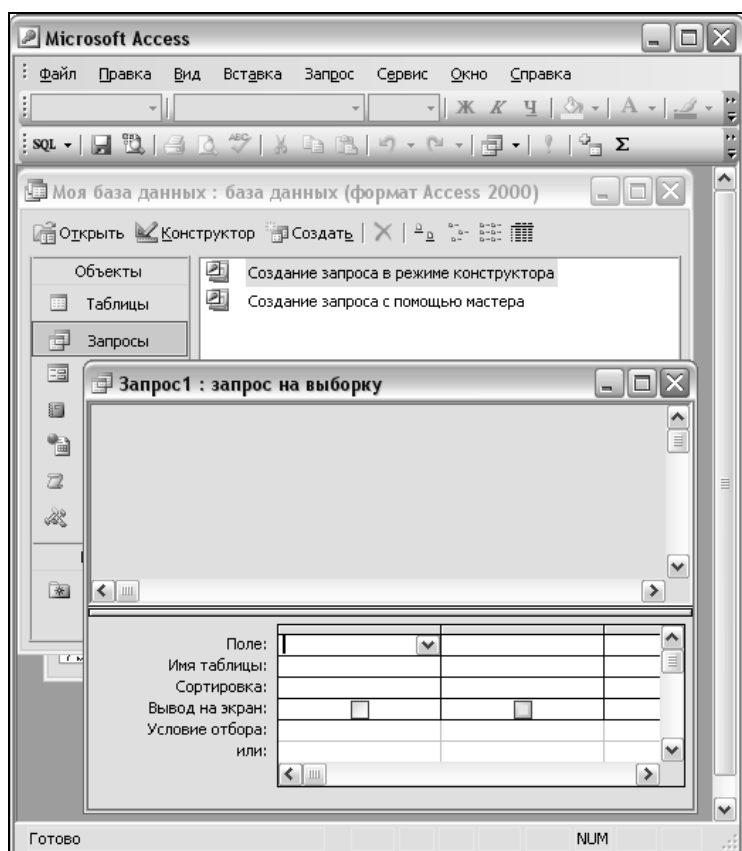


Рис. В7. Окно для ввода SQL-выражений в режиме конструктора

В меню **Вид** главного окна Access выберите команду **Режим SQL** или щелкните правой кнопкой мыши на верхней области окна **Запрос1: запрос на выборку** и в раскрывшемся контекстном меню выберите **Режим SQL**. В результате окно запросов примет вид, как показано на рис. В8. Это текстовый редактор, в котором можно вводить и редактировать выражения на языке SQL. При создании нового запроса в области ввода появляется строка "SELECT;", т. е. ключевое слово запроса, на выборку данных. Если вам необходимо выбрать данные, то необходимо дописать недостающие элементы SQL-выражения. Для создания других запросов (например, модификации данных, создания таблиц и др.) следует удалить ключевое слово SELECT и написать вместо него требуемый оператор.

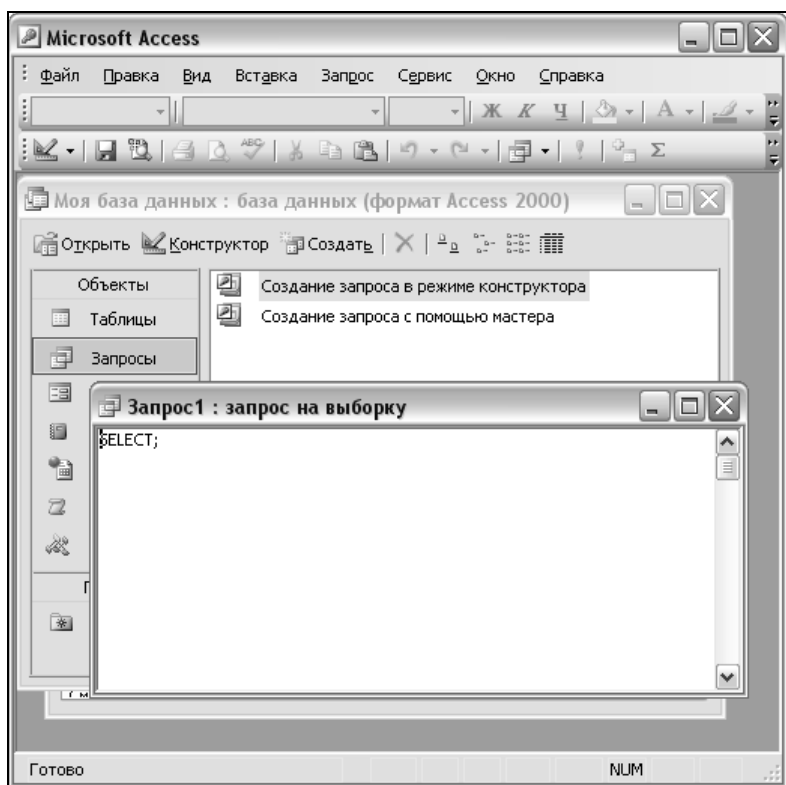


Рис. В8. Окно для ввода SQL-выражений в режиме SQL