

# СОДЕРЖАНИЕ

---

<b>Введение</b> .....	11
<b>Глава 1. Основные характеристики СУБД</b> .....	24
Выполнение программ SQL .....	25
Microsoft Access .....	27
Microsoft SQL Server .....	30
Oracle .....	33
MySQL .....	36
PostgreSQL .....	38
<b>Глава 2. Реляционная модель</b> .....	41
Таблицы, столбцы и строки .....	43
Первичные ключи .....	47
Внешние ключи .....	51
Связи .....	54
Нормализация .....	57
Наша типовая база данных .....	63
<b>Глава 3. Основы SQL</b> .....	69
Синтаксис SQL .....	69
Типы данных .....	75
Строковые типы данных .....	77
Битовые типы данных .....	79
Точные числовые типы данных .....	80
Действительные числовые типы данных .....	82
Календарные типы данных .....	84
Интервальные типы данных .....	88
Значение null .....	90

---

<b>Глава 4. Выбор данных из произвольной таблицы</b> .....	92
Выбор столбцов с помощью предложений SELECT и FROM .....	93
Создание псевдонимов столбцов с помощью предложения AS .....	96
Удаление повторяющихся строк с помощью ключевого слова DISTINCT .....	99
Сортировка строк с помощью предложения ORDER BY .....	101
Фильтрация строк с помощью предложения WHERE .....	109
Комбинирование условий с помощью операторов AND, OR и NOT ...	114
Сравнение по шаблону оператором LIKE .....	122
Сравнение с диапазоном с помощью оператора BETWEEN .....	128
Фильтрация с помощью оператора IN .....	131
Проверка на значение null с помощью оператора IS NULL .....	134
<b>Глава 5. Операторы и функции</b> .....	137
Создание производных столбцов .....	138
Арифметические операции .....	140
Определение последовательности вычисления .....	143
Объединение строк с помощью оператора    .....	145
Выбор произвольной подстроки с помощью функции SUBSTRING() .....	149
Переключение регистра символов строки с использованием функций UPPER() и LOWER() .....	153
Удаление пробелов с помощью функции TRIM() .....	155
Определение длины произвольной строки с помощью функции CHARACTER_LENGTH() .....	159
Поиск подстроки с использованием функции POSITION() .....	161
Операции с данными даты и времени .....	165
Извлечение значений текущих даты и времени .....	169
Отображение информации о пользователе .....	171
Преобразование типов данных с помощью функции CAST() .....	173
Вычисление условных значений с помощью выражения CASE .....	179
Проверка на значения null с использованием функции COALESCE() .....	184
Сравнение выражений с помощью функции NULLIF() .....	185
<b>Глава 6. Суммирование и группировка данных</b> .....	188
Использование агрегатных функций .....	189
Поиск минимума посредством функции MIN() .....	192
Поиск максимума с использованием функции MAX() .....	194
Вычисление суммы с помощью функции SUM() .....	196
Порядок расчета среднего значения с помощью функции AVG() ...	198
Подсчет строк с помощью функции COUNT() .....	200

Исключение повторных значений с помощью предложения DISTINCT .....	202
Группирование строк с использованием предложения GROUP BY .....	206
Фильтрация групп с помощью предложения HAVING .....	215
<b>Глава 7. Выбор данных из нескольких таблиц .....</b>	<b>219</b>
Уточнение имен столбцов .....	220
Создание псевдонимов таблиц с помощью предложения AS .....	222
Использование объединений .....	224
Создание объединений с помощью синтаксиса JOIN и WHERE .....	228
Создание произвольного перекрестного объединения с использованием предложения CROSS JOIN .....	233
Создание произвольного естественного объединения с использованием предложения NATURAL JOIN .....	236
Создание внутреннего объединения с помощью команды INNER JOIN .....	241
Создание внешних объединений с помощью команды OUTER JOIN .....	265
Создание самообъединения .....	279
Комбинирование строк с помощью оператора UNION .....	286
Поиск общих строк с помощью команды INTERSECT .....	295
Поиск разных строк с помощью команды EXCEPT .....	297
<b>Глава 8. Подзапросы .....</b>	<b>299</b>
Принципы работы с подзапросами .....	300
Структура подзапросов .....	302
Подзапросы и объединения .....	303
Простые и сложные запросы .....	307
Определение названий столбцов в подзапросах .....	313
Значения null в подзапросах .....	314
Использование подзапросов в качестве выражений в списке заголовков столбцов .....	316
Сравнение значений, возвращаемых подзапросом, с использованием операторов сравнения .....	320
Проверка на входжение во множество с помощью оператора IN ...	325
Сравнение всех значений запроса с помощью ключевого слова ALL .....	333
Сравнение некоторых значений запроса с помощью ключевого слова ANY .....	337
Проверка наличия выборки с помощью оператора EXISTS .....	341
Сравнение эквивалентных запросов .....	347

<b>Глава 9. Добавление, обновление и удаление строк</b> .....	348
Отображение названий столбцов в таблице .....	349
Вставка строк с помощью команды INSERT .....	352
Изменение строк с помощью команды UPDATE .....	358
Удаление строк с помощью команды DELETE .....	363
<b>Глава 10. Создание, изменение и удаление таблиц</b> .....	367
Порядок создания таблиц .....	368
Основные принципы работы с ограничениями .....	369
Создание новой таблицы с помощью команды CREATE TABLE .....	371
Запрет значения null с помощью ограничения NOT NULL .....	373
Присвоение значения по умолчанию с помощью ограничения DEFAULT .....	376
Задание первичного ключа с помощью ограничения PRIMARY KEY .....	379
Задание внешнего ключа с помощью ограничения FOREIGN KEY .....	383
Присвоение уникальных значений с помощью ограничения UNIQUE .....	389
Проверка значений столбца с помощью ограничения CHECK .....	392
Создание временной таблицы с помощью команды CREATE TEMPORARY TABLE .....	395
Создание новой таблицы на основе существующей с помощью команды SELECT INTO .....	398
Изменение таблицы с помощью команды ALTER TABLE .....	401
Удаление таблицы с помощью команды DROP TABLE .....	404
<b>Глава 11. Индексы</b> .....	405
Создание индекса с помощью команды CREATE INDEX .....	405
Удаление индекса с помощью команды DROP INDEX .....	409
<b>Глава 12. Представления</b> .....	411
Создание представления с помощью команды CREATE VIEW .....	411
Считывание данных из представления .....	417
Изменение данных через представление .....	419
Удаление представления с помощью команды DROP VIEW .....	424
<b>Глава 13. Транзакции</b> .....	425
Выполнение транзакций .....	426
<b>Приложение</b> .....	431
<b>Предметный указатель</b> .....	445

---

*Посвящается моему отцу*

## **Благодарности**

Беки Морган (Becky Morgan) – за критику первого варианта рукописи.

Марджори Баер (Marjorie Baer) – за то, что посоветовала прочитать книгу «Do you know SQL?».

Кэти Симпсон (Kathy Simpson) – за то, что иногда ради работы жертвовала сном.

Лизу Бразиил (Lisa Brazieal) – за постоянное «вырезать и вклеить».

Морин Форис (Maureen Forsys) – за отсутствие переносов в ключевых словах.

Брайена Штайнвега (Bryan Steinweg) – за переносы на следующую страницу.

Даррена Пеннингтона (Darren Pennington) – за то, что сажился на своего любимого «конька».

Нэнси Олдридж-Рюнцель (Nancy Aldrich-Ruenzel) – за свежие идеи.

Информация в примере с базой данных – вымышленная. Я позаимствовал пару названий из новелл Яна М. Бэнкса (Iain M. Bank's).

# ВВЕДЕНИЕ

---

SQL – это стандартный язык программирования, применяемый для создания, модификации, поиска и извлечения информации, хранящейся в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).

С помощью SQL вы можете, в частности, превратить любой вопрос типа «А где живут наши клиенты?» в такую команду, которую программное обеспечение вашей базы данных сможет понять и выполнить (для приведенного вопроса это может быть команда `SELECT city, state FROM customers`). Если вы уже умеете извлекать информацию аналогичного типа с помощью графического инструментария построения запросов, то, скорее всего, заметили, что он становится весьма ограничивающим и громоздким по мере того, как сложность ваших запросов возрастает. Вот здесь и нужен SQL, хотя решением указанной проблемы его возможности не ограничиваются. Например, вы можете применять SQL для того, чтобы добавлять, модифицировать и удалять данные и объекты базы данных. И именно потому, что язык SQL такой мощный, его поддерживают наиболее популярные СУБД, в частности Microsoft Access, Oracle и MySQL, хотя уровень

этой поддержки существенно зависит от того, о какой именно СУБД идет речь.

Прежде чем перейти к собственно программированию, имеет смысл познакомиться с основными идеями реляционных баз данных и SQL. Так мы и поступим.

## SQL

SQL – это:

- язык программирования;
- простой в изучении;
- непроцедурный;
- встроенный и/или интерактивный;
- стандартизованный;
- используемый для манипулирования данными и объектами баз данных;
- не аббревиатура.

Рассмотрим пункты этого списка подробнее.

### Язык программирования

SQL – один из формальных языков, то есть средство, с помощью которого вы передаете компьютеру инструкции, называемые программой. Программное обеспечение вашей базы данных выполняет

---

эту программу, написанную на языке SQL. Это значит, что СУБД выполняет те запросы, которые вы ей передали, и отображает результаты их работы, в том числе какое-нибудь сообщение об ошибке. Надо сказать, что языки программирования, называемые также формальными языками, отличаются от языков общения, называемых неформальными или естественными языками, главным образом тем, что создаются под конкретную цель, полностью лишены двусмысленности, имеют весьма ограниченные словарный запас и гибкость. Таким образом, если вы не получили результата от работы своей программы, на который рассчитывали при ее написании, это произошло потому, что ваша программа содержит какую-либо ошибку (логическую или синтаксическую – в последнем случае, скорее всего, будет выведено соответствующее сообщение, описывающее ошибку), а не потому, что компьютер неправильно понял ваши инструкции, формализованные в виде программы (эта информация проясняет, почему отладка программ считается одной из основных задач программирования).

Будучи формальным языком, SQL, как и другой язык этого типа, имеет свои синтаксис и семантику. *Синтаксис* включает собственно слова и символы, которые вы можете применять, а также правила, по которым эти слова и символы можно использовать при создании команд и программ. *Семантика* помогает выяснить реальное значение, смысл любой синтаксически правильной команды. Вы вполне можете написать на SQL какую-нибудь команду, соответствующую синтаксису языка, которая, тем не менее, будет выражать неверный смысл (то есть будет правильной синтаксически, но неверной семантически). Подробнее о синтаксисе и семантике SQL рассказывается в главе 3.

## Простой в изучении

Уточним: SQL прост в изучении по сравнению с другими языками программирования. Дело в том, что, если вы пока не написали ни одной программы, переход от неформального языка к любому формальному языку вас поначалу сильно разочарует. Тем не менее подчеркнем, что команды на SQL читаются как предложения неформального языка, что уже сильно облегчает понимание смысла. Так, любой новичок в программировании, скорее всего, поймет, что записанная на SQL команда `SELECT au_fname, au_lname FROM authors ORDER BY au_lname` совпадает по смыслу с предложением «Перечислить имена и фамилии всех авторов, сортируя их по фамилиям». Но этот человек почти наверняка сочтет эквивалентную по смыслу программу, написанную на С или на Perl, абсолютно непонятной.

## Непроцедурный

Заметим сразу: если вы никогда не программировали, то можете пропустить этот раздел, не беспокоясь об утрате целостности восприятия остального материала; но если вы программировали на С или Perl, то имели дело с процедурным языком. А для любого процедурного языка программист должен явно прописывать шаги, которые выполняет компьютер, чтобы получить желаемый результат. SQL относится к непроцедурным языкам (такие языки еще называют «декларативными»). Программируя на таком языке, вам надо описать то, что именно вы хотите сделать, а не то, как вы собираетесь это делать. В случае с SQL оптимизатор, который входит в состав программного обеспечения вашей СУБД, самостоятельно рассчитает это самое «как». Именно поэтому язык SQL не содержит таких конструкций

**Листинг 1.** Этот код на Microsoft Access Visual Basic извлекает имена и фамилии из таблицы в базе данных, содержащей информацию об авторах, и помещает результат выборки в выходной массив

Листинг

```
Sub GetAuthorNames()
    Dim db As Database
    Dim rs As Recordset
    Dim i As Integer
    Dim au_names() As String
    Set db = CurrentDb()
    Set rs = db.OpenRecordset("authors")
    rs.MoveLast
    ReDim au_names(rs.RecordCount - 1, 1)
    With rs
        .MoveFirst
        i = 0
        Do Until .EOF
            au_names(i,0) = ![au_fname]
            au_names(i,1) = ![au_lname]
            i = i + 1
            .MoveNext
        Loop
    End With
    rs.Close
    db.Close
End Sub
```

**Листинг 2.** Эта команда SQL выполняет ту же самую задачу, что и вся процедура на VB, показанная в листинге 1. Внутренний оптимизатор СУБД сам определяет, как наилучшим образом извлечь соответствующие данные

Листинг

```
SELECT au_fname, au_lname
FROM authors;
```

управления, как IF-THEN-ELSE, WHILE и операторов перехода GOTO<sup>1</sup>.

Чтобы наглядно продемонстрировать различие между процедурными и непроцедурными языками, мы написали две эквивалентные программы, выполняющие по сути одну и ту же задачу. Первая программа, текст которой приведен в листинге 1, написана на Microsoft Access Visual Basic, то есть на процедурном языке. Вторая программа, текст которой приведен в листинге 2, написана на SQL. Программа на VB извлекает имена авторов из таблицы, содержащей информацию о них. Конечно, нет необходимости вникать в детали первой программы, но обратите внимание, что она использует цикл Do Until для того, чтобы явно определить то, как именно извлекать нужные данные. Вторая программа делает то же самое, но одной-единственной командой SQL. В первом случае мы имеем примерно 20 строк кода, а во втором – всего одну команду. Но важнее всего другое: программируя на SQL, вам надо указать только то, что именно необходимо сделать, а дальше сама СУБД автоматически и незаметно для вас определяет и исполняет ту последовательность пошаговых операций, которую требуется выполнить для достижения желаемого результата. Более того, листинг 2 – это простейший из всех возможных запрос SQL. А если к этому запросу добавить обычные операции сортировки, фильтрации и объединения, то для выполнения всего, что будет делать усложненная, но единственная команда SELECT, записанная на SQL, может понадобиться до 100 строк процедурного кода.

<sup>1</sup> Однако управляющие конструкции обычно присутствуют в SQL и применяются при создании так называемых хранимых процедур – определенных наборов инструкций, предназначенных для выполнения сложных действий с данными. – *Прим. науч. ред.*

## Встроенный и/или интерактивный

Если вы включаете команды SQL как элементы в какие-нибудь программы, написанные на процедурных языках, считается, что вы применяете так называемый встроенный SQL, а те процедурные языки, на которых написаны эти более крупные или, точнее, несущие программы, называют базовыми. Таким языком на практике чаще всего оказывается какой-нибудь язык программирования общего назначения, в частности C, Java или COBOL, или какой-либо язык сценариев, например Perl, PHP или Python. То есть любой сценарий PHP, запускаемый как CGI-программа, может применять вложенные команды SQL, чтобы запрашивать произвольную базу СУБД MySQL. СУБД в этом случае передаст результат этого запроса какой-нибудь переменной PHP для отображения и дальнейшей обработки. Например, как это показано в листинге 3, команда SQL встроена в программу, написанную на Access Visual Basic.

Если в режиме реального времени адресовать команды на языке SQL непосредственно вашей СУБД, а она отобразит соответствующие результаты сразу же, как только получит, значит, вы применяете интерактивный или динамический SQL. Отметим, что все современные серверы СУБД поставляются в комплекте с такими графическими приложениями или утилитами командной строки, которые воспринимают интерактивные команды SQL или/и текстовые файлы, содержащие SQL-программы, то есть сценарии.

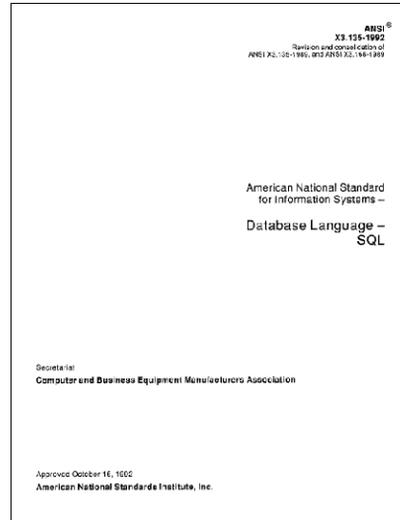
В настоящем издании рассматриваются только интерактивные команды SQL, поскольку их можно встроить в любую прикладную программу или сценарий. Однако имейте в виду, что между интерактивными и встроенными командами могут существовать небольшие синтаксические различия.

**Листинг 3.** Здесь Visual Basic выполняет функцию базового языка для встроенного SQL. Однако во всех примерах данной книги используется синтаксис только интерактивного SQL, который может слегка отличаться от синтаксиса встроенного SQL в зависимости от конкретной СУБД

```

Листинг
Sub GetAuthorNames2()
  Dim db As Database
  Dim rs As Recordset
  Set db = CurrentDb()
  Set rs = db.OpenRecordset("SELECT
→ au_fname, au_lname FROM authors;")
  ' Далее программа обрабатывает результаты
→ запроса, находящиеся в переменной rs.
  rs.Close
  db.Close
End Sub

```



**Рис. 1.** Титульный лист 626-страничного документа ANSI, который называется X3.135-1992 Database Language – SQL и, собственно, официально определяет стандарт SQL-92.

При желании вы можете купить электронную версию этого документа на сайте [www.ansi.org](http://www.ansi.org), но он рассчитан главным образом на тех программистов, кто самостоятельно разрабатывает, пишет и внедряет СУБД

## Стандартизованный

SQL никому не принадлежит. Это значит, что на права собственности никто не претендует. Как легко понять из рис. 1, SQL – это открытый стандарт, разрабатываемый одним из комитетов в Американском национальном институте стандартизации (ANSI). В свою очередь, ANSI – это не частная фирма, а орган правительства США, призванный способствовать развитию национальных стандартов. Более того, SQL является не только американским, но и международным стандартом ISO/IEC 9075:1992, так как в 1992 году был включен в число стандартов Международной организации по стандартизации (ISO). В этой связи хотелось бы заметить, что настоящее издание основано на стандарте ANSI SQL 1992. Таким образом, когда вы встретите обозначения ANSI SQL, SQL-92 или просто SQL, их следует рассматривать как синонимы, если специально не указано ничего другого. Положение несколько усложняется тем, что появился стандарт ANSI SQL-99. Однако на него пока можно не обращать внимания, так как до его внедрения пройдет еще несколько лет и он будет обратно совместим со стандартом SQL-92. Значит, программа, отвечающая стандарту SQL-92, будет работать под любой СУБД, отвечающей стандарту SQL-99.

Следует заметить, что все поставщики СУБД в разной степени привносят свои дополнения в канонический ANSI SQL, чтобы сделать его более мощным. Чаще всего такими расширениями являются дополнительные команды, ключевые слова, функции, типы данных и конструкции управляющей логики, например IF, WHILE и FOR. Иногда, правда, этих расширений оказывается слишком много. В частности, Oracle и Microsoft уже внесли так много добавлений, что появились PL/SQL и Transact-SQL – самостоятельные

полноправные языки, а не надмножества SQL. Однако обычно расширения одного поставщика несовместимы с расширениями другого. Имея это в виду, мы указываем читателю на те элементы «расширенного» SQL, которые не соответствуют стандарту ANSI (см. раздел «Выполнение команд SQL на коммерческих СУБД»).

## Используемый для манипулирования данными и объектами базы данных

Все команды SQL принято делить на две основные группы – язык манипулирования данными (DML) и язык определения данных (DDL).

Для любой базы данных команды группы DML отбирают, обсчитывают, вставляют, удаляют и редактируют те данные, которые хранятся в этой базе. В настоящем издании рассматриваются команды SELECT, INSERT, UPDATE и DELETE (см. главы 4–9).

Команды группы DDL создают, модифицируют и уничтожают такие объекты базы данных, как таблицы, индексы и представления. В нашей книге речь идет о командах CREATE, ALTER и DROP (см. главы 10–13).

## Несколько слов о произношении

SQL нельзя произносить как «СИКВЭЛ» (английское слово *sequel* означает «последствие» или «продолжение»). Избегайте этой ошибки, а лучше четко проговаривайте каждую букву из трех: S-Q-L. Мы не согласны с теми людьми, которые произносят «СИКВЭЛ» только потому, что так говорят многие. Дело в том, что смешение *sequel* и SQL является историческим курьезом, образованные люди знают об этом, и для них вариант «СИКВЭЛ» будет свидетельствовать о таком же низком уровне владения ремеслом, как если бы вы оказались писателем и при написании

своих романов стали бы разбивать инфинитивы, вставляя наречия после частицы to (что тоже в английском языке вроде бы допускается, но в серьезной литературе почти никогда не встречается). Короче говоря, помните, что «СИКВЭЛ» неприятно звучит для уха профессионала. А еще MySQL следует произносить как «МАЙ-ЭС-КЬЮ-ЭЛЬ», а PostgreSQL – как «ПОСТ-ГРЕС-КЬЮ-ЭЛЬ».

### Немного о названии

К сожалению, довольно распространено еще одно заблуждение относительно SQL. Многие считают, что сочетание трех букв S-Q-L есть аббревиатура, которая расшифровывается как Structured Query Language (Структурированный язык запросов). Так вот, SQL означает SQL, и более ничего. Почему? Да потому, что так заявлено в ANSI. Официальное название SQL – Database Language SQL (Язык баз данных SQL). Обращаем особое внимание на данное заблуждение только потому, что оно вовсе не так безобидно. Если начинающие программисты будут считать, что SQL и вправду означает «язык структурированных запросов», это сослужит им плохую службу, так как именно этот язык является наихудшим из всех возможных описаний канонического SQL, хотя живучесть данного описания кажется академикам и профессионалам забавной по следующим причинам:

- SQL – не структурированный язык, поскольку его нельзя разбить на блоки или процедуры;
- SQL не ограничивается только запросами, поскольку в нем есть много других команд помимо SELECT;
- SQL не является полным языком по определению Тюринга.

## Несколько общих замечаний о книге

Эта книга рассказывает об использовании SQL для сохранения в базе данных и запрашивания из нее нужной информации. В главах с 1 по 3 приводится общий описательный материал по СУБД, по реляционной модели и по синтаксису SQL. Начиная с главы 4 изложение становится более наглядным и основывается на конкретных примерах.

Автор, безусловно, рассчитывал на то, что читатель хорошо разбирается в своей ОС (будь то Windows, Unix, Mac OS или др.), в частности понимает, как работает файловая система, умеет создавать, редактировать, удалять и организовывать файлы, папки и каталоги. В некоторых задачах требуется умение вводить команды по приглашению операционной системы или командной оболочки (в старых версиях Windows его еще называют приглашением DOS).

Изложение материала книги ограничено наиболее ходовыми командами ANSI SQL, поэтому ее нельзя назвать исчерпывающим руководством. Для изучения команд, которые здесь не рассматриваются, рекомендуем обзавестись подробными справочными пособиями.

### Сайт поддержки

На сайте [www.peachpit.com/vqs/sql](http://www.peachpit.com/vqs/sql) вы найдете все материалы книги (с исправленными ошибками). Свои вопросы, предложения и замечания высылайте по электронной почте: [fehil@pacbell.net](mailto:fehil@pacbell.net).

## Для кого предназначена эта книга

Эта книга – для вас, если вы:

- не обладаете большим опытом в программировании, но умеете работать на компьютере;
- изучаете SQL самостоятельно или с инструктором;
- не интересуетесь базами данных, но по роду своей деятельности вынуждены обрабатывать большие объемы структурированной информации (как это часто имеет место в работе бухгалтера, научного сотрудника, Web-программиста, аналитика рынка, торгового представителя, финансового советника, руководителя офиса и даже секретаря);
- хотите выйти за ограничения, налагаемые дружественным, но крайне мало мощным графическим инструментарием построения запросов по образцу (QBE – Query by Example);
- намерены перейти от программного обеспечения настольной СУБД к серверному программному обеспечению, поддерживающему SQL;
- уже знакомы с SQL и хотите узнать о нем как можно больше;
- должны по роду деятельности создавать, модифицировать или/и удалять такие объекты баз данных, как таблицы, индексы и представления;
- вынуждены встраивать SQL в программы, написанные на языках Java, C или Perl;
- являетесь Web-программистом и работаете с MySQL или PostgreSQL;
- просто хотите иметь настольный справочник по SQL.

Эта книга вам совершенно не подходит, если вы планируете изучить следующие вопросы:

- как проектировать базы данных (хотя обзоры толковых идей, которыми следует руководствоваться, приводится в главе 2);
- частные расширения, которые поставщики СУБД добавили к ANSI SQL;
- как профессионально и с применением современных методов программировать и администрировать базы данных (мы не касаемся, например, инсталляции, триггеров, хранимых процедур, тиражирования, резервирования, восстановления, оптимизации, курсоров, схем сортировки, сравнений, алфавитов и трансляции).

## Дополнительные соглашения

Иногда части текста выделяются разными шрифтами:

- моноширинный шрифт служит для написания кода как в листингах, так и в тексте книги и для обозначения экранного текста в командной строке;
- *курсив* применяется при вводе новых терминов и, совместно с моноширинным шрифтом, при обозначении заменяемых имен идентификаторов;
- участки рисунков, на которые следует обратить внимание, обведены;
- полужирным моноширинным шрифтом отмечены результаты, о которых говорится в тексте;
- **полужирным шрифтом** выделены элементы интерфейса.

При рассмотрении схем синтаксиса SQL и листингов будем придерживаться следующих правил:

- для улучшения читабельности и сохранности текста кода служит плотная печать. А для экономии места на страницах книги ширина отступа (величина отступа слева при начале абзаца) выбрана равной двум пробелам, хотя обычно она равна четырем пробелам;

### Настольные и серверные СУБД, поддерживающие SQL

Любая серверная СУБД, поддерживающая SQL, работает в качестве серверной части архитектуры «клиент-сервер». Это значит, что она хранит и поддерживает базы данных и отвечает на запросы, которые клиенты оформляют на языке SQL. Здесь под клиентом понимается любое приложение или любой компьютер, которые могут посылать запросы на языке SQL (в дальнейшем мы будем называть такие запросы SQL-запросами) на сервер и получать ответы с этого сервера. Например, если в вашей локальной сети применяется архитектура «клиент-сервер», то клиентом является компьютер на вашем рабочем столе, а сервером – мощная специальная машина, расположенная в соседней комнате, в соседнем здании или в другой стране. При этом правила, описывающие порядок передачи запросов и соответствующих ответов в архитектуре «клиент-сервер», определяются протоколом связи с СУБД, например ODBC и JDBC.

Любая настольная СУБД – это СУБД, установленная и работающая локально. Она представляет собой независимое приложение, которое хранит свои собственные базы данных и производит всю обработку данных, связанную с SQL. Никакая настольная СУБД не может принимать запросы от других клиентов, что, конечно, означает, что никакая настольная СУБД не может работать так, как это делает любой SQL-сервер.

В качестве примеров SQL-сервера можно привести Microsoft SQL Server, Oracle, MySQL и PostgreSQL (заметим, что понятие SQL-сервера включает в себя понятие серверной СУБД, поддерживающей SQL, но не совпадает с ним). А в качестве настольных СУБД, поддерживающих SQL, можно назвать Microsoft Access и FileMaker Pro. В отношении программных продуктов, англоязычные названия которых содержат словосочетание SQL server, следует знать об одном нюансе. Дело в том, что SQL server (слово server написано с маленькой буквы) относится к коммерческому серверному продукту, поддерживающему SQL, произвольного поставщика, а SQL Server (слово Server написано с большой буквы) – только к конкретному продукту SQL Server компании Microsoft.

В соответствии с устоявшимися требованиями мы будем использовать слова «клиент» («клиентский») и «сервер» («серверный») по отношению не только к клиентскому и, соответственно, серверному программному обеспечению, но и к той машине, на которой работает соответствующее программное обеспечение. В тех случаях, когда надо подчеркнуть различие между компьютером и программным обеспечением, мы будем делать специальные пояснения.

- каждая команда SQL начинается с новой строки;
- как вы узнаете из главы 3, SQL является языком свободной формы без ограничений на разрыв строки или на число слов в одной строке. У нас же каждое предложение любой команды начинается с новой строки и с отступом от первой строки, например:

```
SELECT au_fname, au_lname
FROM authors
ORDER BY au_lname;
```

- в SQL регистр не учитывается, следовательно, идентификаторы `myname`, `MyName` и `MYNAME` считаются идентичными с точки зрения канонического SQL. Однако верхний регистр используется нами для выделения ключевых слов, таких как `SELECT`, `NULL` и `CHARACTER` (см. раздел «Синтаксис SQL» в главе 3), а нижний регистр – для выделения элементов, конкретные значения которых должны быть определены пользователем, как это имеет место, например, для имен таблиц, столбцов и для псевдонимов. Стоит особо подчеркнуть, что в некоторых СУБД имена идентификаторов, определяемые пользователем, являются, в отличие от канонического SQL, чувствительными к регистру. Поэтому безопаснее все-таки учитывать регистр при написании программ;
- в табл. 1 поясняется смысл специальных символов, используемых в схемах и комментариях синтаксиса SQL;
- все кавычки в кодах SQL только прямые – или одинарные (`'`), или двойные (`"`). Никакие другие кавычки (имеются в виду фигурные) недопустимы. Неправильные кавычки не воспринимаются транслятором, и код, содержащий их, работать не будет;
- когда колонка, отведенная для строки кода, окажется для нее слишком узкой, то строка будет разбита на две или более

частей. Стрелка ( $\rightarrow$ ) в начале строки означает, что эта строка является продолжением предыдущей.

С

Этим значком помечены советы, которые помогут вам в практической работе.

П

Таким образом оформлены примечания, содержащие дополнительную информацию по теме раздела.

## Выполнение команд SQL на коммерческих СУБД



В тексте книги иногда встречается пиктограмма, которая означает, что речь идет об отступлении производителями СУБД от канонов стандартного ANSI SQL-92. Поэтому, увидев такой значок, вам придется определенным образом модифицировать код, приведенный в каком-нибудь листинге, под авторскую версию SQL конкретного поставщика СУБД, чтобы вы смогли запустить этот код на своей СУБД. Например, если в тексте говорится, что оператором объединения (конкатенации) двух строк в ANSI SQL является `||` (двойной конвейер или труба), но продукты Microsoft применяют для этой цели знак `«+»` (плюс), а MySQL применяет вместо оператора конкатенации функцию `CONCAT()`, вам, если вы работаете на соответствующей СУБД, придется во всех выражениях типа *allb* в запросе поменять знак `«||»` или на знак `«+»`, или на функцию `CONCAT()`.

В подавляющем большинстве случаев наши программы SQL будут работать или совсем без изменений, или с незначительными синтаксическими поправками на всех основных СУБД. К сожалению, хотя и редко, некоторые программы не будут работать совсем. Скажем, MySQL 4.0 и более ранних версий не поддерживает

подзапросы (хотя их поддержка планировалась в версии 4.0).

Перечислим СУБД, учитываемые в настоящем издании:

- Microsoft Access 2002;
- Microsoft SQL Server 2000;
- Oracle Release 9i;
- MySQL 4.0;
- PostgreSQL 7.1.

Мы остановились на этих СУБД, потому что они самые ходовые. Все программы SQL, приведенные в данной книге, были протестированы указанными коммерческими версиями СУБД. Поскольку соответствие базовому ANSI SQL в последующих версиях никогда не ухудшается, наши программы будут работать и на всех будущих версиях этих СУБД.

Сразу заметим, программы из этой книги вы можете запустить и на DB2, и на Sybase Adaptive Server, и на Informix. Если какая-то программа откажется работать, прочитайте документацию по соответствующей СУБД и определите, в чем ее реализация SQL отличается от стандартного ANSI SQL.

**П**

В список избранных СУБД версия Oracle 8i тоже включена. Просто имейте в виду: если номер версии этой СУБД не указан, значит, примечания или советы имеют отношение как к Oracle 9i, так и к Oracle 8i.

**Таблица 1.** Специальные символы, применяемые для объяснений правил синтаксиса

Знаки	Назначение
	Символ «вертикальная черта» или «труба» разделяет альтернативные элементы, выбор одного из которых может быть, судя по контексту, как обязательным, так и необязательным. В любом случае можно выбрать не более одного из этих элементов. Никогда не печатайте этот символ в тексте программ! Выражение <code>A B C</code> будет означать или «Или А, или В, или С», или «Или А, или В, или С, или ничего». Не путайте знак трубы со знаком двойной трубы <code>  </code> . Дело в том, что двойная труба является оператором SQL и применяется для конкатенации (объединения) строк
[]	В квадратные скобки заключаются элементы, выбор хотя бы одного из которых необязателен. Никогда не печатайте этот символ в тексте программ! Выражение <code>[A B C]</code> будет означать: «Печатай или А, или В, или С или не печатай ничего»
{}	В фигурные скобки заключаются элементы, выбор одного из которых обязателен. Никогда не печатайте этот символ в тексте программ! Выражение <code>{A B C}</code> будет означать: «Печатай или А, или В, или С»
...	Многоточие означает, что предыдущий элемент/элементы можно повторить любое количество раз
()	В отличие от всех предыдущих знаков, круглые скобки являются элементом языка SQL. Их надо печатать там, где указывает соответствующая грамматическая схема

### Разница между базой данных и СУБД

Любая база данных – это совсем не то программное обеспечение, на котором вы работаете. Поэтому говорить, что Oracle – это база данных, неправильно. Полное название программного обеспечения, обслуживающего базы данных, звучит так: система управления базами данных (СУБД). Важно понять, что любая база данных является не более чем компонентом некой СУБД, что она является элементом данных, будучи не более чем контейнером (состоящим из одного или нескольких файлов), содержащим структурированную информацию. Помимо задач контроля, организации, выборки, поддержания целостности и достоверности данных любая СУБД решает еще задачи обеспечения физического хранения, безопасности, тиражирования данных и исправления ошибок при их обработке.

Иногда вместо аббревиатуры СУБД используют аббревиатуру РСУБД, имея в виду именно реляционную СУБД. Такая СУБД организует данные в соответствии с реляционной моделью (о реляционной модели мы подробнее поговорим в главе 2). Хотим отметить, что в настоящем издании рассматриваются только реляционные базы данных, поэтому при использовании аббревиатуры СУБД речь идет о РСУБД.

### Что такое FileMaker Pro

Если говорить кратко, FileMaker Pro – это распространенное настольное приложение со встроенной базой данных, которое поддерживает команду SQL SELECT с предложениями FROM, WHERE и ORDER BY (подробнее об этом см. в главе 4). Этих возможностей вполне достаточно для выполнения самых распространенных типов запросов. Чтобы запустить любой допустимый SQL-запрос на FileMaker Pro, можно воспользоваться инструментарием SQL Query Builder или оператором Execute SQL script. Основательно разобраться в этих вопросах поможет справочная система FileMaker Pro или книга Цинтии Л. Барон и Даниеля Пека (Cynthia L. Baron, Daniel Peck) «FileMaker Pro 5/5.5 Advanced: Visual QuickPro Guide». Наконец, вы можете многое узнать о FileMaker Pro на сайте [www.filemaker.com](http://www.filemaker.com).

## Программное обеспечение, необходимое для полноценной работы с этой книгой

Чтобы воспроизвести примеры настоящей книги на компьютере, вам понадобится:

- текстовый редактор;
- типовая база данных;
- СУБД.

### Текстовый редактор

Хотя печатать короткие интерактивные команды SQL в ответ на приглашение командной строки может показаться даже удобным, обычно вы все-таки стараетесь сохранять нетривиальные запросы в текстовых файлах. В соответствии с общепринятыми правилами имена файлов SQL имеют расширение `.sql`, хотя вы можете выбрать любое другое расширение, например `.txt`.

Любой текстовый редактор – это некая служебная программа для создания и модификации текстовых файлов. Как известно, такие файлы содержат только печатные буквы, числа и символы, но никаких шрифтов, никакого форматирования, никаких «невидимых» знаков, цветов и графики, то есть ничего из той системы, которую принято связывать с понятием текстового процессора, в них нет. Хотя каждая операционная система обычно поставляется вместе с неким текстовым редактором (см. табл. 2), вы вполне можете поэкспериментировать с разными коммерческими и условно бесплатными текстовыми редакторами (см. сайт [www.download.com](http://www.download.com)).

Таблица 2. Распространенные текстовые редакторы

Среда	Текстовый редактор
Windows	Notepad
Приглашение командной строки	
Windows	edit
Unix, Linux	vi, emacs, pico
Mac OS	Teach Text, Simple Text
Mac OS X Terminal	vi, emacs, pico

**П** Все программы SQL, содержащиеся в настоящей книге, можно скачать с сайта поддержки (см. раздел «Несколько общих замечаний о книге»).

**П** Вы вполне можете обойтись и без текстового редактора, используя какой-нибудь текстовый процессор, скажем Microsoft Word, и сохраняя соответствующие файлы как «text only». Но такой подход весьма проблематичен, и профессионалы рассматривают его как плохой стиль.

## Типовая база данных

Большинство примеров, рассматриваемых в данной книге, рассчитаны на одну и ту же базу данных (см. раздел «Наша типовая база данных» в главе 2). Чтобы ее построить, запустите SQL-скрипт, приведенный в приложении, или, если вы работаете с Microsoft Access, откройте файл books.mdb (все эти файлы вы можете скачать с сайта поддержки). Если вы работаете с действующей серверной СУБД организации, то на создание базы данных и на запуск команд SQL вам может понадобиться официальное разрешение системного администратора.

## Система управления базами данных

Для выполнения SQL-запросов просто передайте их какой-нибудь СУБД, которая «понимает» SQL. И уже она выполнит ваш запрос и отобразит соответствующие результаты. В первой главе мы расскажем о том, как запускать SQL-запросы на самых распространенных СУБД.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)