# Contents

Introduction.................................................................................................................1

1 **Understanding SQL** .............................................................................................................. 5
   Database Basics .............................................................................................................. 5
   What Is SQL? ................................................................................................................. 11
   Try It Yourself .............................................................................................................. 12
   Summary ...................................................................................................................... 12

2 **Retrieving Data** ............................................................................................................ 13
   The **SELECT** Statement .......................................................................................... 13
   Retrieving Individual Columns ................................................................................. 14
   Retrieving Multiple Columns .................................................................................... 16
   Retrieving All Columns ............................................................................................... 17
   Summary ...................................................................................................................... 18

3 **Sorting Retrieved Data** .............................................................................................. 19
   Sorting Data .................................................................................................................. 19
   Sorting by Multiple Columns ....................................................................................... 21
   Sorting by Column Position ....................................................................................... 22
   Specifying Sort Direction ............................................................................................ 23
   Summary ...................................................................................................................... 25

4 **Filtering Data** ............................................................................................................... 26
   Using the **WHERE** Clause ....................................................................................... 26
   The **WHERE** Clause Operators ............................................................................ 28
   Summary ...................................................................................................................... 32

5 **Advanced Data Filtering** ............................................................................................ 33
   Combining **WHERE** Clauses .................................................................................... 33
   Using the **IN** Operator ............................................................................................. 36
   Using the **NOT** Operator ........................................................................................ 38
   Summary ...................................................................................................................... 39
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Using Wildcard Filtering</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Using the LIKE Operator</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>Tips for Using Wildcards</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>47</td>
</tr>
<tr>
<td>7</td>
<td>Creating Calculated Fields</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Understanding Calculated Fields</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>Concatenating Fields</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Performing Mathematical Calculations</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>Using Data Manipulation Functions</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Understanding Functions</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Using Functions</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>65</td>
</tr>
<tr>
<td>9</td>
<td>Summarizing Data</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Using Aggregate Functions</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Aggregates on Distinct Values</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Combining Aggregate Functions</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>74</td>
</tr>
<tr>
<td>10</td>
<td>Grouping Data</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Understanding Data Grouping</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Creating Groups</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>Filtering Groups</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Grouping and Sorting</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>SELECT Clause Ordering</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>83</td>
</tr>
<tr>
<td>11</td>
<td>Working with Subqueries</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Understanding Subqueries</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Filtering by Subquery</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Using Subqueries As Calculated Fields</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>90</td>
</tr>
</tbody>
</table>
12 Joining Tables ................................................................. 91
   Understanding Joins ......................................................... 91
   Creating a Join .............................................................. 94
   Summary ........................................................................ 100

13 Creating Advanced Joins .................................................. 101
   Using Table Aliases ......................................................... 101
   Using Different Join Types .............................................. 102
   Using Joins with Aggregate Functions ............................ 108
   Using Joins and Join Conditions ....................................... 110
   Summary ........................................................................ 110

14 Combining Queries .......................................................... 111
   Understanding Combined Queries ................................. 111
   Creating Combined Queries .......................................... 112
   Summary ........................................................................ 117

15 Inserting Data ................................................................. 118
   Understanding Data Insertion .......................................... 118
   Copying from One Table to Another ............................... 124
   Summary ........................................................................ 126

16 Updating and Deleting Data .............................................. 127
   Updating Data .............................................................. 127
   Deleting Data .............................................................. 129
   Guidelines for Updating and Deleting Data ...................... 131
   Summary ........................................................................ 132

17 Creating and Manipulating Tables .................................... 133
   Creating Tables ........................................................... 133
   Updating Tables .......................................................... 139
   Deleting Tables ........................................................... 141
   Renaming Tables .......................................................... 141
   Summary ........................................................................ 142
Using Microsoft Access .................................................................197
Using Microsoft ASP .................................................................199
Using Microsoft ASP.NET .......................................................199
Using Microsoft Query.............................................................200
Using Microsoft SQL Server ..................................................201
Using MySQL............................................................................202
Using Oracle..............................................................................202
Using PHP................................................................................203
Using PostgreSQL....................................................................203
Using SQL Server .....................................................................205
Configuring ODBC Data Sources .....................................205

C  SQL Statement Syntax .................................................................207
    ALTER TABLE ............................................................................207
    COMMIT .................................................................................208
    CREATE INDEX .........................................................................208
    CREATE PROCEDURE .............................................................208
    CREATE TABLE .........................................................................208
    CREATE VIEW ............................................................................209
    DELETE ...................................................................................209
    DROP ..........................................................................................209
    INSERT ....................................................................................209
    INSERT SELECT .........................................................................210
    ROLLBACK .................................................................................210
    SELECT ....................................................................................210
    UPDATE ....................................................................................211

D  Using SQL Datatypes .................................................................212
    String Datatypes .................................................................213
    Numeric Datatypes .............................................................215
    Date and Time Datatypes ....................................................216
    Binary Datatypes .................................................................217

E  SQL Reserved Words .................................................................219

Index ..........................................................................................225
Introduction

SQL is the most widely used database language. Whether you are an application developer, database administrator, Web application designer, or Microsoft Office user, a good working knowledge of SQL is an important part of interacting with databases.

This book was born out of necessity. I had been teaching Web application development for several years, and students were constantly asking for SQL book recommendations. There are lots of SQL books out there. Some are actually very good. But they all have one thing in common: for most users they teach just too much information. Instead of teaching SQL itself most books teach everything from database design and normalization to relational database theory and administrative concerns. And while those are all important topics, they are not of interest to most of us who just need to learn SQL.

And so, not finding a single book that I felt comfortable recommending, I turned that classroom experience into the book you are holding. Sams Teach Yourself SQL in 10 Minutes will teach you SQL you need to know, starting with simple data retrieval and working on to more complex topics including the use of joins, subqueries, stored procedures, cursors, triggers, and table constraints. You’ll learn methodically, systematically, and simply—in lessons that will each take 10 minutes or less to complete.

Now in its third edition, this book has taught SQL to hundreds of thousands of users, and now it is your turn. So turn to Lesson 1, and get to work. You’ll be writing world class SQL in no time at all.

Who is the Teach Yourself SQL Book For?

This book is for you if

- You are new to SQL.
- You want to quickly learn how to get the most out of SQL.
• You want to learn how to use SQL in your own application development.

• You want to be productive quickly and easily in SQL without having to call someone for help.

DBMSs Covered in This Book
For the most part, the SQL taught in this book will apply to any Database Management System (DBMS). However, as all SQL implementations are not created equal, the following DBMSs are explicitly covered (and specific instructions or notes are included where needed):

• IBM DB2
• Microsoft Access
• Microsoft SQL Server
• MySQL
• Oracle
• PostgreSQL
• Sybase Adaptive Server

Example databases and SQL scripts are also available for all of these DBMSs.

Conventions Used in This Book
This book uses different typefaces to differentiate between code and regular English, and also to help you identify important concepts.

Text that you type and text that should appear on your screen is presented in monospace type.

It will look like this to mimic the way text looks on your screen.
Placeholders for variables and expressions appear in monospace italic font. You should replace the placeholder with the specific value it represents.

This arrow (➡️) at the beginning of a line of code means that a single line of code is too long to fit on the printed page. Continue typing all the characters after the ➡️ as though they were part of the preceding line.

**A Note** presents interesting pieces of information related to the surrounding discussion.

**A Tip** offers advice or teaches an easier way to do something.

**A Caution** advises you about potential problems and helps you steer clear of disaster.

New Term icons provide clear definitions of new, essential terms.

**INPUT** The Input icon identifies code that you can type in yourself.

**OUTPUT** The Output icon highlights the output produced by running a program.

**ANALYSIS** The Analysis icon alerts you to the author’s line-by-line analysis of a program.
This page intentionally left blank
In this lesson, you will learn how to use the SELECT statement’s ORDER BY clause to sort retrieved data as needed.

Sorting Data

As you learned in the last lesson, the following SQL statement returns a single column from a database table. But look at the output. The data appears to be displayed in no particular order at all.

```
SELECT prod_name
FROM Products;
```

```
prod_name
--------------------
Fish bean bag toy
Bird bean bag toy
Rabbit bean bag toy
8 inch teddy bear
12 inch teddy bear
18 inch teddy bear
Raggedy Ann
King doll
Queen doll
```

Actually, the retrieved data is not displayed in a mere random order. If unsorted, data will typically be displayed in the order in which it appears in the underlying tables. This could be the order in which the data was added to the tables initially. However, if data was subsequently updated or deleted, the order will be affected by how the DBMS reuses reclaimed
storage space. The end result is that you cannot (and should not) rely on the sort order if you do not explicitly control it. Relational database design theory states that the sequence of retrieved data cannot be assumed to have significance if ordering was not explicitly specified.

Clause SQL statements are made up of clauses, some required and some optional. A clause usually consists of a keyword and supplied data. An example of this is the SELECT statement’s FROM clause, which you saw in the last lesson.

To explicitly sort data retrieved using a SELECT statement, the ORDER BY clause is used. ORDER BY takes the name of one or more columns by which to sort the output. Look at the following example:

```
INPUT
SELECT prod_name
FROM Products
ORDER BY prod_name;
```

```
ANALYSIS
This statement is identical to the earlier statement, except it also specifies an ORDER BY clause instructing the Database Management System software to sort the data alphabetically by the prod_name column. The results are as follows:
```

```
OUTPUT
prod_name
----------------------
12 inch teddy bear
18 inch teddy bear
8 inch teddy bear
Bird bean bag toy
Fish bean bag toy
King doll
Queen doll
Rabbit bean bag toy
Raggedy Ann
```

```
Position of ORDER BY Clause When specifying an ORDER BY clause, be sure that it is the last clause in your SELECT statement. Using clauses out of order will generate an error message.
```
Sorting by Nonselected Columns  
More often than not, the columns used in an `ORDER BY` clause will be ones that were selected for display. However, this is actually not required, and it is perfectly legal to sort data by a column that is not retrieved.

## Sorting by Multiple Columns

It is often necessary to sort data by more than one column. For example, if you are displaying an employee list, you might want to display it sorted by last name and first name (first by last name, and then within each last name sort by first name). This would be useful if there are multiple employees with the same last name.

To sort by multiple columns, simply specify the column names separated by commas (just as you do when you are selecting multiple columns).

The following code retrieves three columns and sorts the results by two of them—first by price and then by name.

```
SELECT prod_id, prod_price, prod_name
FROM Products
ORDER BY prod_price, prod_name;
```

<table>
<thead>
<tr>
<th>prod_id</th>
<th>prod_price</th>
<th>prod_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNBG02</td>
<td>3.4900</td>
<td>Bird bean bag toy</td>
</tr>
<tr>
<td>BNBG01</td>
<td>3.4900</td>
<td>Fish bean bag toy</td>
</tr>
<tr>
<td>BNBG03</td>
<td>3.4900</td>
<td>Rabbit bean bag toy</td>
</tr>
<tr>
<td>RGAN01</td>
<td>4.9900</td>
<td>Raggedy Ann</td>
</tr>
<tr>
<td>BR01</td>
<td>5.9900</td>
<td>8 inch teddy bear</td>
</tr>
<tr>
<td>BR02</td>
<td>8.9900</td>
<td>12 inch teddy bear</td>
</tr>
<tr>
<td>RYL01</td>
<td>9.4900</td>
<td>King doll</td>
</tr>
<tr>
<td>RYL02</td>
<td>9.4900</td>
<td>Queen doll</td>
</tr>
<tr>
<td>BR03</td>
<td>11.9900</td>
<td>18 inch teddy bear</td>
</tr>
</tbody>
</table>

It is important to understand that when you are sorting by multiple columns, the sort sequence is exactly as specified. In other words, using the output in the example above, the products are sorted by the `prod_name` column only when multiple rows have the same `prod_price` value. If all
the values in the `prod_price` column had been unique, no data would have been sorted by `prod_name`.

### Sorting by Column Position

In addition to being able to specify sort order using column names, `ORDER BY` also supports ordering specified by relative column position. The best way to understand this is to look at an example:

**Input**
```
SELECT prod_id, prod_price, prod_name
FROM Products
ORDER BY 2, 3;
```

**Output**
```
<table>
<thead>
<tr>
<th>prod_id</th>
<th>prod_price</th>
<th>prod_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNBG02</td>
<td>3.4900</td>
<td>Bird bean bag toy</td>
</tr>
<tr>
<td>BNBG01</td>
<td>3.4900</td>
<td>Fish bean bag toy</td>
</tr>
<tr>
<td>BNBG03</td>
<td>3.4900</td>
<td>Rabbit bean bag toy</td>
</tr>
<tr>
<td>RGAN01</td>
<td>4.9900</td>
<td>Raggedy Ann</td>
</tr>
<tr>
<td>BR01</td>
<td>5.9900</td>
<td>8 inch teddy bear</td>
</tr>
<tr>
<td>BR02</td>
<td>8.9900</td>
<td>12 inch teddy bear</td>
</tr>
<tr>
<td>RYL01</td>
<td>9.4900</td>
<td>King doll</td>
</tr>
<tr>
<td>RYL02</td>
<td>9.4900</td>
<td>Queen doll</td>
</tr>
<tr>
<td>BR03</td>
<td>11.9900</td>
<td>18 inch teddy bear</td>
</tr>
</tbody>
</table>
```

As you can see, the output is identical to that of the query above. The difference here is in the `ORDER BY` clause. Instead of specifying column names, the relative positions of selected columns in the `SELECT` list are specified. `ORDER BY 2` means sort by the second column in the `SELECT` list, the `prod_price` column. `ORDER BY 2, 3` means sort by `prod_price` and then by `prod_name`.

The primary advantage of this technique is that it saves retyping the column names. But there are some downsides too. First, not explicitly listing column names increases the likelihood of you mistakenly specifying the wrong column. Second, it is all too easy to mistakenly reorder data when making changes to the `SELECT` list (forgetting to make the corresponding changes to the `ORDER BY` clause). And finally, obviously you cannot use this technique when sorting by columns that are not in the `SELECT` list.
Sorting Retrieved Data

Specifying Sort Direction

Data sorting is not limited to ascending sort orders (from A to Z). Although this is the default sort order, the ORDER BY clause can also be used to sort in descending order (from Z to A). To sort by descending order, the keyword DESC must be specified.

The following example sorts the products by price in descending order (most expensive first):

```
SELECT prod_id, prod_price, prod_name
FROM Products
ORDER BY prod_price DESC;
```

```
prod_id    prod_price    prod_name
-------    ----------    --------------------
BR03       11.9900       18 inch teddy bear
RYL01      9.4900        King doll
RYL02      9.4900        Queen doll
BR02       8.9900        12 inch teddy bear
BR01       5.9900        8 inch teddy bear
RGAN01     4.9900        Raggedy Ann
BNBG01     3.4900        Fish bean bag toy
BNBG02     3.4900        Bird bean bag toy
BNBG03     3.4900        Rabbit bean bag toy
```

But what if you were to sort by multiple columns? The following example sorts the products in descending order (most expensive first), plus product name:

```
SELECT prod_id, prod_price, prod_name
FROM Products
ORDER BY prod_price DESC, prod_name;
```
### OUTPUT

<table>
<thead>
<tr>
<th>prod_id</th>
<th>prod_price</th>
<th>prod_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR03</td>
<td>11.9900</td>
<td>18 inch teddy bear</td>
</tr>
<tr>
<td>RYL01</td>
<td>9.4900</td>
<td>King doll</td>
</tr>
<tr>
<td>RYL02</td>
<td>9.4900</td>
<td>Queen doll</td>
</tr>
<tr>
<td>BR02</td>
<td>8.9900</td>
<td>12 inch teddy bear</td>
</tr>
<tr>
<td>BR01</td>
<td>5.9900</td>
<td>8 inch teddy bear</td>
</tr>
<tr>
<td>RGAN01</td>
<td>4.9900</td>
<td>Raggedy Ann</td>
</tr>
<tr>
<td>BNBG02</td>
<td>3.4900</td>
<td>Bird bean bag toy</td>
</tr>
<tr>
<td>BNBG01</td>
<td>3.4900</td>
<td>Fish bean bag toy</td>
</tr>
<tr>
<td>BNBG03</td>
<td>3.4900</td>
<td>Rabbit bean bag toy</td>
</tr>
</tbody>
</table>

### Analysis

The `DESC` keyword only applies to the column name that directly precedes it. In the example above, `DESC` was specified for the `prod_price` column, but not for the `prod_name` column. Therefore, the `prod_price` column is sorted in descending order, but the `prod_name` column (within each price) is still sorted in standard ascending order.

#### Sorting Descending on Multiple Columns

If you want to sort descending on multiple columns, be sure each column has its own `DESC` keyword.

It is worth noting that `DESC` is short for `DESCENDING`, and both keywords may be used. The opposite of `DESC` is `ASC` (or `ASCENDING`), which may be specified to sort in ascending order. In practice, however, `ASC` is not usually used because ascending order is the default sequence (and is assumed if neither `ASC` nor `DESC` are specified).
Case Sensitivity and Sort Orders  When you are sorting textual data, is A the same as a? And does a come before B or after Z? These are not theoretical questions, and the answers depend on how the database is set up.

In dictionary sort order, A is treated the same as a, and that is the default behavior for most Database Management Systems. However, most good DBMSs enable database administrators to change this behavior if needed. (If your database contains lots of foreign language characters, this might become necessary.)

The key here is that if you do need an alternate sort order, you cannot accomplish it with a simple ORDER BY clause. You must contact your database administrator.

Summary
In this lesson, you learned how to sort retrieved data using the SELECT statement’s ORDER BY clause. This clause, which must be the last in the SELECT statement, can be used to sort data on one or more columns as needed.
INDEX

SYMBOLS
% (percent sign) wildcard, 41-42
[] (square brackets) wildcard, 44-46
^ (caret) wildcard character, 45
_ (underscore) wildcard, 43-44
| (pipe) symbol, 207
\ (backslash) concatenation operator, 50
'(single quotation mark), WHERE clause operators and, 30
* (asterisk) wildcard, queries, 17-18
*= (equality operator), 107
+ (concatenation operator), 50
+ (plus sign operator), outer joins, 107
, (comma), multiple column separation, 16
@ (at symbol) character, 158
@@ERROR variable, 167
@@IDENTITY global variable, 160

A
ABS( ) function, 64
Access (Microsoft)
  DISTINCT argument support, 72
equivalent tables for, 192
  pass-through mode, 198
  running, 197-198
  sorting by alias, 82
  stored procedure support, 154
aggregate functions
  ALL argument, 72
  AVG( ), 67-68
  combining, 73-74
  COUNT( ), 68-69
  defined, 67
  DISTINCT argument, 72-73
  outer joins and, 108-110
  MAX( ), 69
  MIN( ), 70
  naming aliases, 74
  overview, 66
  SUM( ), 71
aliases
  alternative uses, 54
  columns, creating, 101
  concatenating fields, 53-54
  names, 54
aggregate functions and
  74
  self joins, 102-104
  table names, 101-102
ALL argument, aggregate functions, 72
ALL clause, grouping data, 77
alphabetical sort order, 23-24
ALTER TABLE statements, 139-140
CHECK constraints, 180
CONSTRAINT syntax, 177
syntax, 207
AND keyword, 31
AND operator, 33-34
ANSI SQL, 12
applications
filtering query results, 27
SQL compatibility
Aqua Data Studio, 195
ColdFusion (Macromedia), 196
DB2 (IBM), 195-196
Microsoft Access, 197-198
Microsoft ASP, 199
Microsoft ASP.NET, 199-200
Microsoft Query, 200-201
Microsoft SQL Server, 201
MySQL, 202
ODBC configuration, 203
Oracle, 202
PHP scripting language, 203
PostgreSQL, 203
Query Tool, 203-204
selection criteria, 194
Sybase Adaptive Server, 204-205
Aqua Data Studio
running, 195
Web site, 194
arguments
ALL, aggregate functions, 72
DBMS support, 73
DISTINCT, aggregate functions, 72-73
AS keyword, 53-54
Oracle support, 102
ASC keyword, query results sort order, 24
ASP (Microsoft), running, 199
ASP.NET (Microsoft), running, 199-200
authentication, 185
authorization, 185
AVG( ) function, 67-68
DISTINCT argument, 72
NULL values, 68
B
BETWEEN operator, 63
WHERE clause, 30
between specified values operator (WHERE clause), 28
BINARY datatype, 217
BIT datatype, 215
C
calculated fields
concatenating fields, 49-52
column aliases, 53-54
mathematical calculations, 55-56
overview, 48-49
subqueries, 88-90
views, 151-152
calculated values, totaling, 71
Cartesian Product, joins and, 95-97
cascading deletes, 178
case sensitivity
query result sort order, 25
SQL statements, 16
CHAR string datatype, 214
characters, searching for
% (percent sign) wildcard, 41-42
[ ] (square brackets) wildcard, 44-46
_ (underscore) wildcard, 43-44
check constraints, 179-180
clauses, 20
ALL, grouping data, 77
GROUP BY, 76-77
HAVING, grouping data, 78
INDEX

IS NULL, 31
SELECT statements, order of, 83
WHERE, 26
  AND operator, 33-34
  checking against single value, 29
  checking for nonmatches, 29-30
  checking for NULL value, 31
  checking for range of values, 30-31
  IN operator, 36-38
  joins and, 95-97
  multiple query criteria, 33
  NOT operator, 38-39
  operator support by DBMS, 30
  operators, 28
  OR operator, 34-35
  order of evaluation, 35-36
  positioning, 27
  SOUNDEX function, 61
client-based results formatting, compared to server-based, 49
CLOSE statements, closing cursors, 173
code (programming)
  commenting, 59
  stored procedures, 159
  portability, 58
ColdFusion (Macromedia), running, 196
columns. See also fields
  aliases
    alternative uses, 54
    concatenating fields, 53-54
    creating, 101
    names, 54
  AVG( ) function, individual columns, 68
  breaking data correctly, 8
  concepts, 8
  Customers example table, 189
derived, 54
fully qualified names, 95
GROUP BY clause, 77
grouping data, specifying by relative position, 77
Identity fields, 160
INSERT SELECT statements, 124
INSERT statement, omitting columns, 122
insert STATEMENT AND, 120
multiple, sorting query results by, 21-22
nonselected, sorting query results by, 23
NULL value, checking for, 31
NULL value columns, 136-137
OrderItems example table, 191
Orders example table, 190
  padded spaces, RTRIM( ) function, 51-52
position, sorting query results by, 22
primary keys, 10
Products example table, 189
retrieving
  all, 17-18
  individual, 14-15
  multiple, 16-17
  unknown, 18
  separating names in queries, 16
  sorting data, descending on multiple columns, 24
  subquery result restrictions, 88
  updating multiple, 128
  values, deleting, 129
  Vendors example table, 188
combined queries
  creating, 112-113
  duplicate rows and, 114-115
  overview, 111
  performance, 114
  rules, 114
  sorting results, 116

How can we make this index more useful? Email us at indexes@samspublishing.com
Command Center utility, running, 195-196
commas (,), multiple column separation, 16
commenting
  programming code, importance of, 59
  stored procedure code, 159
COMMIT statement (transaction processing), 165
  commits defined, 163
  syntax, 208
compatibility
datatypes, 9
  functions, DBMS support considerations, 57-58
  WHERE clause operators, 28
compatibility (SQL code), application selection criteria, 194
concatenating fields, 49-52
  column aliases, 53-54
  mathematical calculations, 55-56
  MySQL, 51
concatenation operators, 50
configuring ODBC, 205-206
CONSTRAINT syntax, ALTER TABLE statements, 177
constraints (referential integrity)
  check constraints, 179-180
  foreign keys, 176-177
  overview, 174-175
  primary keys, 175-176
  speed, 184
  unique constraints, 178-179
copying tables, 126
COS( ) function, 64
COUNT( ) function, 67-69
  DISTINCT argument, 73
  joins and, 108
  NULL values, 69
COUNT* subquery, 89
CREATE INDEX statement, 182
  syntax, 208
CREATE TABLE statement, 133-135
  DEFAULT keyword, 137-138
  syntax, 208
CREATE VIEW statement, 146
  syntax, 209
cross joins, 97
currency datatypes, 216
cursors
  accessing, 171-173
  closing, 173
  creating, 170-171
  implementing, 170
  limitations, 169
  opening, 171
  options, support for, 168
  overview, 168
  Web-based applications, 169
Customers table, 189

D
data
  breaking correctly (columns), 8
deleting
    guidelines, 131
    TRUNCATE TABLE statement, 131
filtering, indexes and, 182
manipulation functions, date and time, 63
security, 185
updating, guidelines, 131
Database Management System. See DBMS
databases. See also tables
  concepts, 5-6
defined, 6
dropping objects, 209
indexes
  cautions, 182
  creating, 182
  scalability, 93
  schemas, 7
security, 185
tables
creating, 208
triggers, 183
DATABASELENGTH( ) function, 60
datatypes, 8
binary, 217
compatibility, 9
currency, 216
data and time, 216
defining, 180
numeric, 215
string, 213
usefulness of, 212
BIT, 215
CHAR, 217
compatibility, 9
currency, 216
DATE, 216
DATETIME, 216
DBMS differences, 213
DECIMAL, 215
defining, 180
INT, 215
NCHAR, 214
NVARCHAR, 214
TEXT, 214
TINYINT, 215
user-defined, 180
VARBINARY, 217
date (system), default value syntax, 138
date and time datatypes, 216
date and time functions, 59, 62-64
DATE datatype, 216
DATEPART( ) function, 62
DATETIME datatype, 216
DB2 (IBM), running, 195-196
DBMS (Database Management System), 6
accidental table deletion, 141
datatype differences, 213
functions, support considerations, 57-58
indexes, 182
interactive tools, 93
ISAM databases, 169
LIKE operator, search patterns and, 41
NULL value differences, 137
query sort order, 20
security mechanisms, 185
SQL extensions, 12
transaction processing, implementation differences, 164
triggers, 184
TRIM functions, 52
UNION statements, 117
user-defined datatypes, 180
view creation, 144
views, rules and restrictions, 145
WHERE clause, allowed operators, 32
DECIMAL datatype, 215
DECLARE statements
cursors, creating, 170-171
stored procedures, 158
default values, tables, 137-138
defining datatypes, 180
DELETE FROM statements, 130
DELETE statement, 129-130
FROM keyword, 130
guidelines, 131
security privileges, 130
syntax, 209
transaction processing, 163
TRUNCATE TABLE statement, 131
DELETE statements
rollbacks, 165
triggers, 183
WHERE clause, 130
deleting
column values, 129
data
guidelines, 131
TRUNCATE TABLE statement, 131
How can we make this index more useful? Email us at indexes@samspublishing.com
rows, 209
preventing accidental, 178
tables, 141
preventing accidental, 141
derived columns. See aliases
DESC keyword, query results sort order, 23-24
dictionary sort order (query results), 25
DISTINCT argument
AVG( ) function, 72
COUNT( ) function, 73
double pipes (||), concatenation operator, 50
downloading example tables, 191
Microsoft Access MDB file, 192
SQL scripts, 192
DROP statement, syntax, 209
DROP TABLE statement, 141
dropping database objects, 209

E
empty strings, compared to NULL values, 137
equality ( *= ) operator, 107
equality operator (WHERE clause), 28
establishing primary keys, 10
example tables
Customers table, 189
downloading, 191
functions of, 187
Microsoft Access MDB file, 192
OrderItems table, 191
Orders table, 190
Products table, 189
SQL scripts, 192
Vendors table, 188
EXCEPT statement, 117
EXECUTE statement, stored procedures, 156-160
EXP( ) function, 64
explicit commits, 165
extensions, 12

F
FETCH statement, accessing cursors, 171-173
fields. See also calculated fields;
columns
aliases, names, 54
calculated
concatenating fields, 49-54
mathematical calculations, 55-56
overview, 48-49
subqueries, 88-90
views, 151-152
filtering
% (percent sign) wildcard, 41-42
[ ] (square brackets) wildcard, 44-46
_ (underscore) wildcard, 43-44
data, indexes, 182
data groups, 78-80
LIKE operator, 40-41
query results, 26
AND operator, 33-34
application level, 27
IN operator, 36-38
multiple criteria, 33
NOT operator, 38-39
OR operator, 34-35
order of evaluation, 35-36
WHERE clause operators, 28-31
by subqueries, 84-88
with views, 150
fixed length strings, 213
FLOAT datatype, 215
foreign keys, 176-177
greater than or equal to operator  
(WHERE clause), 28
GROUP BY clause, 76-77
compared to ORDER BY clause, 80-82
grouping
data, 75
  columns, specifying by position, 77
  compared to sorting, 80-82
  filtering groups, 78-80
  GROUP BY clause, 76-77
  nested groups, 76
operators, 35

transaction processing, 163
triggers, 183
VALUES, 121
INT datatype, 215
integrity. See referential integrity
interactive DBMS tools, 93
INTERSECT statements, 117
INTO keyword, 119
IS NULL clause, 31
ISAM (Indexed Sequential Access Method) databases, 169
DISTINCT argument, aggregate functions, 72-73

HAVING clause, grouping data, 78
IBM DB2, running, 195-196
Identity fields, 160
IN operator, 36-38
indexes
  cautions, 182
  creating, 182, 208
  overview, 180-182
  revisiting, 182
inner joins, 97-98
INSERT SELECT statement, 122-124
  SELECT INTO statement comparison, 125
  syntax, 210
INSERT statement
  columns lists, 121
  completing rows, 118-120
  INTO keyword, 119
  omitting columns, 122
  overview, 118
  partial rows, 121-122
  query data, 122-124
  rollbacks, 167
  security privileges, 118
  syntax, 209

key processing, 163
triggers, 183
VALUES, 121
INT datatype, 215
integrity. See referential integrity
interactive DBMS tools, 93
INTERSECT statements, 117
INTO keyword, 119
IS NULL clause, 31
ISAM (Indexed Sequential Access Method) databases, 169
DISTINCT argument, aggregate functions, 72-73

joins
aggregate functions and,
  108-110
Cartesian Product, 95-97
creating, 94
cross joins, 97
inner joins, 97-98
multiple tables, 98-100
natural joins, 104-105
outer, 105-108
  full, 108
  left, 107
  right, 107
  syntax, 107
  types, 108
overview, 91
performance considerations, 99
self joins, 102-104
usefulness of, 93
views, 147-148
WHERE clause, 95-97

keys, primary, 9-11
keywords, 13
  AND, 31, 34
  AS, 53-54
  Oracle support, 102
ASC, query results sort order, 24
DEFAULT, table values, 137-138
DESC, query results sort order, 23-24
FROM, 14, 129
IN, 38
INTO, 119
NOT, 38
OR, 35
REFERENCES, 177
UNIQUE, 179

L
languages, SQL, 11
LCASE( ) function, 60
LEFT keyword (outer joins), 106
left outer joins, 107
LEFT( ) function, 60
LEN( ) function, 60
LENGTH( ) function, 60
less than operator (WHERE clause), 28
less than or equal to operator (WHERE clause), 28
LIKE operator, 40-41
% (percent sign) wildcard, 41-42
[] (square brackets), 44-46
_ (underscore) wildcard, 43-44
local variables, @ character, 158
logical operators, defined, 33
LONG RAW datatype, 217
LOWER( ) function, 60
LTRIM( ) function, 52, 60

M
Macromedia ColdFusion, running, 196
mathematical
  calculations, 55-56
  operators, 56
MAX( ) function, 67-69
  DISTINCT argument, 73
  non-numeric data, 70
  NULL values, 70
Microsoft Access
  DISTINCT argument support, 72
  example tables for, 192
  pass-through mode, 198
  running, 197-198
  sorting by alias, 82
  stored procedure support, 154
Microsoft ASP, running, 199
Microsoft ASP.NET, running, 199-200
Microsoft Query, running, 200-201
Microsoft SQL Server, running, 201
MIN( ) function, 67, 70
  DISTINCT argument, 73
  non-numeric data, 70
  NULL values, 71
MySQL
  concatenation, 51
  cursor support, 168
  NOT operator, 39
  running, 202
  stored procedure support, 154
  subquery support, 84
  views, support for, 143

N
naming
  aliases, 54
    aggregate functions and, 74
  columns, fully qualified names, 95
  indexes, 182
  tables, 7
    reserved words and, 13
    aliases, 101-102
natural joins, 104-105
navigating tables, cursors, 168
NCHAR string datatype, 214
nested data groups, 76
non-equality operator (WHERE clause), 28
non-numeric data
   MAX( ) function, 70
   MIN( ) function, 70
not greater than operator (WHERE clause), 28
not less than operator (WHERE clause), 28
NOT operator, 38-39
   character searching and, 46
NULL keyword, updating columns, 129
NULL value operator (WHERE clause), 28
NULL values
   AVG( ) function, 68
   checking for, 31
   compared to empty strings, 137
   COUNT( ) function, 69
   MAX( ) function, 70
   MIN( ) function, 71
   primary keys, 137
   SUM( ) function, 72
   table columns, 136-137
numeric
datatypes, 215
functions, 59, 64-65
values, quotes, 215
NVARCHAR string datatype, 214
operators
   *= (equality), 107
   % (percent sign) wildcard, 41-42
   + (plus sign), outer joins, 107
   [ ] (square brackets) wildcard, 44-46
   _ (underscore) wildcard, 43-44
   AND, 33-34
   BETWEEN, 63
   concatenation, 50
   defined, 33
   grouping related, 35
   HAVING clause, 78
   IN, 36-38
   LIKE, 40-41
   mathematical, 56
   NOT, 38-39
   OR, 34-35
   order of evaluation, 35-36
   predicates, 41
WHERE clause, 28
   checking against single value, 29
   checking for nonmatches, 29-30
   checking for NULL value, 31
   checking for range of values, 30-31
   compatibility, 28
OR operator, 34-35
Oracle
   commits, 165
cursors
   closing, 173
   creating, 170
   retrieving data, 171
date and time manipulation
   functions, 63
date formatting, 64
   running, 202
   savepoints, 166
   stored procedures, 157
   triggers, 184
ORDER BY clause (SELECT statement), 20
  ascending/descending sort order, 23-24
  compared to GROUP BY clause, 80-82
  positioning, 20
  sorting by column position, 22
  sorting by multiple columns, 21-22
  sorting by nonselected columns, 23
OrderItems table, 191
Orders table, 190
outer joins, 105-108
  full, 108
  left, 107
  right, 107
  syntax, 106-107
  types, 108
overwriting tables, 135

P
parentheses, multiple query criteria
  order of evaluation, 35
pass-through mode (Microsoft Access), 198
patterns (searching), wildcards, 40-41
  % (percent sign), 41-42
  [] (square brackets), 44-46
  _ (underscore), 43-44
percent sign (%) wildcard, 41-42
performance
  combining queries, 114
  deleting data, 131
  indexes, 181
  joins and, 99
  subqueries, 88
  views, 145
PHP scripting language, running, 203
PI( ) function, 65
pipe (|) symbol, 207
placeholders. See savepoints
plus sign (+)
  concatenation operator, 50
  outer joins, 107
portability
  defined, 58
  INSERT statements and, 121
PostgreSQL
  filter query data, 27
  running, 203
predicates (operators), 41
primary keys, 175-176
  concepts, 9-11
  Customer example table, 190
  importance, 10
  NULL values, 137
OrderItems example table, 191
Orders example table, 190
Products example table, 189
Vendors example table, 188
processing
  subqueries, 86
  transactions, 161
Products table, 189
programming code
  commenting, 59
  portability, 58

Q
queries
  calculated fields
    concatenating fields, 49-54
    mathematical calculations, 55-56
    overview, 48-49
  combined
    creating, 112-113
    duplicate rows and, 114-115
    overview, 111
    performance, 114
    rules, 114

How can we make this index more useful? Email us at indexes@samspublishing.com
sorting results, 116
WHERE clauses, 111
combining, 86
data formatting, 17
defined, 84
filtering results, 26
  AND operator, 33-34
  IN operator, 36-38
  multiple criteria, 33
  NOT operator, 38-39
  OR operator, 34-35
  order of evaluation, 35-36
WHERE clause operators, 28-31
INSERT statement and,
  122-124
multiple WHERE clauses, 113
result sets, 168
sorting results, 19-20
  ascending/descending
  order, 23-24
  by column position, 22
  by multiple columns, 21-22
  by nonselected columns,
  21, 23
  case sensitivity, 25
subqueries
  as calculated fields, 88-90
  filtering by, 84-88
  overview, 84
  processing, 86
  self joins and, 103
table aliases, 102
unsorted results, 15
views, 144
wild cards (*), 17-18
Query (Microsoft), running, 200-201
Query Tool
  running, 203-204
Web site, 194
quotation marks
  numeric values, 215
  single (‘), 30
  string values, 214
R
RAW datatype, 217
REAL datatype, 215
records, compared to rows, 9
REFERENCES keyword, 177
referential integrity
  cascading deletes, 178
  constraints
    check constraints, 179-180
    foreign keys, 176-177
    overview, 174-175
    primary keys, 175-176
    unique constraints, 178-179
  natural joins, 104-105
  outer joins, 105-108
  self joins, 102-104
reformatting retrieved data with
  views, 148-149
relational databases (DBMS)
  nonrelational behavior, inducing, 169
  sort order and, 20
relational tables, 91-92
relationships
  constraints
    check constraints, 179-180
    foreign keys, 176-177
    overview, 174-175
    primary keys, 175-176
    unique constraints, 178-179
  natural joins, 104-105
  outer joins, 105-108
  self joins, 102-104
RENAME statement, 141
renaming tables, 141
reserved words, 13, 219
  list of, 220-224
restrictions, views, 145-146
result sets, 168
reusable views, creating, 148
revisiting indexes, 182
REVOKE statements, 185
RIGHT keyword (outer joins), 106	right outer joins, 107
RIGHT( ) function, 60
ROLLBACK command (transaction processing), 164-165
ROLLBACK statement, syntax, 210
rollbacks (transaction processing), 163
  COMMIT statement, 165
defined, 163
ROLLBACK command, 164-165
savebacks and, 166-167
statements, 167
rows
  adding to tables, 209
  compared to records, 9
  concepts, 9
cursors, 168
deleting, 209
INSERT statement, 118-120
  partial rows, 121-122
preventing accidental deletion, 178
updating, 211
RTRIM( ) function, 51-52, 60
rules
  combining queries, 114
  constraints, 175
  views, 145-146
search patterns
  defined, 40
  wildcards, 40-41
    % (percent sign) wildcard, 41-42
    [ ] (square brackets) wildcard, 44-46
    _ (underscore) wildcard, 43-44
cautions, 46
searching
  indexes, overview, 180-182
  trailing spaces and, 43
  wildcards
    ^ (caret) character, 45
    % character, 41-42
    [ ] (square brackets), 44-46
    _ (underscore), 43-44
security
  data, 185
  DELETE statement, 130
  INSERT statements, 118
  UPDATE statement, 127
SELECT INTO statement, 125
  INSERT SELECT statement comparison, 125
SELECT statement, 13
  aggregate functions, combining, 73-74
  AS keyword, 53-54
  AVG( ) function, 67
clauses, ordering of, 83
columns
  retrieving all, 17-18
  retrieving individual, 14-15
  retrieving multiple, 16-17
  retrieving unknown, 18
combining
  creating, 112-113
duplicate rows and, 114-115
overview, 111
rules, 114
sorting results, 116
concatenating fields, 50-51

How can we make this index more useful? Email us at indexes@samspublishing.com
COUNT( ) function, 69
FROM clause, creating joins, 94
grouping data, 76-77
IS NULL clause, 31
ORDER BY clause, 20
positioning, 20
syntax, 210
SELECT statements
subqueries, formatting, 87
WHERE clause, 26
WHERE clauses
combined queries, 111
combining, 33
NOT operator, 38
self joins, 102-104
compared to subqueries, 104
semicolon (;), multiple statements, 16
sequence (SELECT statement clauses), 83
server-based results formatting, compared to client-based, 49
SET command, updating tables, 128
SIN( ) function, 65
single quotation marks (‘)
WHERE clause operators and, 30
SMALLDATETIME datatype, 216
SMALLINT datatype, 215
sorting
combined query results, 116
compared to grouping, 80-82
datatype functionality, 212
indexes, overview, 180-182
query results, 19-20
ascending/descending order, 23-24
by column position, 22
by multiple columns, 21-22
by nonselected columns, 21-23
case sensitivity, 25
SOUNDEX( ) function, 60-61
support for, 61
spaces
removing, RTRIM function, 51-52
search results and, 43
specifying dates, 216
speed, constraints versus triggers, 184
SQL
deleting/updating data, 132
extensions, 12
overview, 11
scripts, example tables, 192
SQL Server
cursors, closing, 173
Identity fields, 160
local variables, @ character, 158
running, 201
savepoints, 166
stored procedures, 158
triggers, 184
SQRT( ) function, 65
square brackets ([]) wildcard, 44-46
statements
ALTER TABLE, 139-140
syntax, 207
case sensitivity, 16
clauses, 20
COMMIT, 165
syntax, 208
CREATE INDEX, 182
syntax, 208
CREATE TABLE, 133-135
syntax, 208
CREATE VIEW, 146
syntax, 209
DELETE, 129-131
FROM keyword, 130
syntax, 209
transaction processing, 163
DROP, syntax, 209
DROP TABLE, 141
formatting, 135
GRANT, 185
grouping related operators, 35
INDEX 239

INSERT
completing rows, 118-120
omitting columns, 122
overview, 118
partial rows, 121-122
query data, 122-124
security privileges, 118
syntax, 209
transaction processing, 163
VALUES, 121
INSERT SELECT, syntax, 210
multiple, separating, 16
OPEN CURSOR, 171
RENAME, 141
REVOKE, 185
ROLLBACK, syntax, 210
rollbacks, 163, 167
defined, 163
SELECT, 13
AVG( ) function, 67
combining, 111-116
combining aggregate functions, 73-74
concatenating fields, 50-51
COUNT( ) function, 69
grouping data, 76-77
retrieving all columns, 17-18
retrieving individual columns, 14-15
retrieving multiple columns, 16-17
retrieving unknown columns, 18
syntax, 210
stored procedures
creating, 157-160
disadvantages of, 155-156
executing, 156-157
overview, 153-154
usefulness of, 154-155
syntax, 207-211
UPDATE, 127-131
syntax, 211
transaction processing, 163
white space, 15
stored procedures
commenting code, 159
creating, 157-160, 208
disadvantages of, 155-156
executing, 156-157
Identity fields, 160
Oracle, 157
overview, 153-154
triggers, 183
usefulness of, 154-155
storing
date and time values, 216
numeric values, cautions, 214
strings, 213
string datatypes, 213
strings
empty, compared to NULL values, 137
fixed length, 213
quotes, 214
TRIM functions, 52
variable-length, 213
wildcard searching and, 41
subqueries
as calculated fields, 88-90
compared to self joins, 104
COUNT*, 89
filtering by, 84-88
formatting, 87
multiple columns, 72
NULL values, 72
overview, 84
performance, 88
processing, 86
self joins and, 103
SUM( ) function, 67, 71
UPDATE statement, 129
WHERE clauses, 88
support, DBMS function support, 57-58

How can we make this index more useful? Email us at indexes@samspublishing.com
Sybase Adaptive Server
running, 204-205
statements, ending, 16
syntax
ALTER TABLE statements, 207
column aliases, 101
COMMIT statement, 208
CREATE INDEX statement, 208
CREATE TABLE statement, 133, 208
CREATE VIEW statement, 209
DELETE statement, 209
DROP statement, 209
INERT statement, 210
INSERT statement, 209
outer joins, 106
ROLLBACK statement, 210
SELECT statement, 210
statements, 207-211
transaction processing, 164
triggers, 184
UPDATE statement, 211
system date, default value syntax, 138
system functions, 59

T

tables
 calculated fields
 concatenating fields, 49-54
 mathematical calculations, 55-56
 overview, 48-49
columns, 8
 aliases, creating, 101
 NULL value, checking for, 31
 primary keys, 10
concepts, 7
copying, 126
copying data into tables, 124-126
creating, 208
 CREATE TABLE statement,
 134-135
 overview, 133-134
datatatypes, 8
default values, 137-138
deleting, 141
 preventing accidental deletion, 141
examples
 Customers table, 189
downloading, 191
indexes
 cautions, 182
creating, 182
searching, 181
INSERT statement, multiple rows, 124
inserting data, 118-120
partial rows, 121-122
from queries, 122-124
joins
 Cartesian Product, 95-97
creating, 94
cross joins, 97
inner joins, 97-98
multiple tables, 98-100
overview, 91
performance considerations, 99
usefulness of, 93
WHERE clause, 95-97
Microsoft Access MDB file, 192
naming, 7
 reserved words and, 13
natural joins, 104-105
NULL value columns, 136-137
outer joins, 105-108
relational, 91-92
renaming, 141
replacing, 135
indexes, 9
adding, 209
deleting, 209
updating, 211
schemas, 7
security, 185
SQL scripts, 192
functions of, 187
OrderItems table, 191
Orders table, 190
Products table, 189
Vendors table, 188
table name aliases, 101-102
self joins, 102-104
triggers, 183
creating, 184
functionality, 183
updating, 127-129, 139-140
deleting data, 129-130
views, creating, 209
tan( ) function, 65
testing, Query Tool and, 203-204
text functions, 59-60
list of common, 60
text string datatype, 214
time functions, 59, 62-64
TINYINT datatype, 215
tools (DBMS), interactive, 93
top argument, 73
top percent argument, 73
totaling values
calculated values, 71
sum( ) function, 71
to_char( ) function, 63
to_number( ) function, 63
transaction processing, 166-167
blocks, rollback statements, 210
commit command, 165
defined, 163
explicit commits, 165
managing, 164
overview, 161-162
rollback command, 164-165
terminology, 163
writing to databases, 208
triggers, 183
creating, 184
functionality, 183
overview, 183-184
speed, 184
syntax examples, 184
trim( ) function, 52
trimming padded spaces, 51-52
troubleshooting
accidental table deletion, 141
Query Tool and, 203-204
 truncated table statement, 131

U
ucase( ) function, 60
underscore (_ ) wildcard, 43-44
union operator
combined queries, 112-113
compared to where clauses, 115
duplicate rows and, 114-115
rules, 114
sorting results, 116
limits, 113
union statements, types, 117
unions (queries)
creating, 112-113
duplicate rows and, 114-115
overview, 111
rules, 114
sorting results, 116
unique constraints, 178-179
unique keyword, 179
unsorted data, query results, 15
update statement, 127-129
from keyword, 129
guidelines, 131
security privileges, 127
subqueries, 129
syntax, 211
transaction processing, 163
triggers, 183

updating
  data, guidelines, 131
tables, 127-129, 139-140
deleting data, 129-130

UPPER( ) function, 59-60
user-defined datatypes, 180

V
values
  concatenation, 50
  searching for (indexes), 181
  trimming padded space, 52
VARBINARY datatype, 217
variable-length strings, 213
Vendors table, 188
views (tables)
  calculated fields, 151-152
    creating, 145
creating, 209
DBMS consistency, 144
filtering data, 150
joins, simplifying, 147-148
overview, 143-146
performance concerns, 145
reformatting retrieved data, 148-149
reusable, 148
rules and restrictions, 145-146
usefulness of, 144-145
virtual tables. See views

W-X-Y-Z
Web sites
  Aqua Data Studio, 194
  example table download site, 191
  Query Tool, 194, 204
  Web-based applications, cursors, 169
WHERE clause, 26. See also HAVING clause
  BETWEEN operator, 30
  combining in queries, 111
  compared to UNION statement, 115
  DELETE statement, 130
  filtering groups, 79
  joins, 97
  joins and, 95-97
  multiple query criteria, 33
    AND operator, 33-34
    IN operator, 36-38
    NOT operator, 38-39
    OR operator, 34-35
    order of evaluation, 35-36
  NOT operator, 38
  operators, 28
    checking against single value, 29
    checking for nonmatches, 29-30
    checking for NULL value, 31
    checking for range of values, 30-31
    quotes and, 30
  operators support by DBMS, 30
  parentheses and, 36
  positioning, 27
  SOUNDEX( ) function, 61
  subqueries, 87
  UPDATE statements, 127-128
  wildcards, 40
white space, SQL statements, 15
wildcards
  asterisk (*) character, 17-18
caret (^) character, 45
cautions, 46
defined, 40
LIKE operator and, 40-41
natural joins, 105
writing stored procedures, 155
YEAR( ) function, 63