This page intentionally left blank
# Table of Contents

**Introduction** 1

- What Is MariaDB Crash Course? 1
- Who Is This Book For? 2
- Companion Web Site 3
- Conventions Used in This Book 3

**1: Understanding SQL** 5

Database Basics 5
- What Is a Database? 6
- Tables 6
- Columns and Datatypes 7
- Rows 8
- NULL 8
- Primary Keys 9
- What Is SQL? 10
- Try It Yourself 11
- Summary 11

**2: Introducing MariaDB** 13

- What Is MariaDB? 13
  - Client-Server Software 14
  - MySQL Compatibility 15
- MariaDB Tools 16
  - mysql Command Line 16
  - MySQL Workbench 17
- Summary 19

**3: Working with MariaDB** 21

- Making the Connection 21
- Selecting a Database 22
- Learning About Databases and Tables 23
- Summary 26

**4: Retrieving Data** 27

- The SELECT Statement 27
- Retrieving Individual Columns 27
- Retrieving Multiple Columns 29
- Retrieving All Columns 30
Retrieving Distinct Rows 31
Limiting Results 32
Using Fully Qualified Table Names 34
Using Comments 35
Summary 36

5: Sorting Retrieved Data 37
Sorting Data 37
Sorting by Multiple Columns 39
Specifying Sort Direction 40
Summary 43

6: Filtering Data 45
Using the \texttt{WHERE} Clause 45
The \texttt{WHERE} Clause Operators 46
\quad Checking Against a Single Value 47
\quad Checking for Nonmatches 48
\quad Checking for a Range of Values 49
\quad Checking for No Value 50
Summary 51

7: Advanced Data Filtering 53
Combining \texttt{WHERE} Clauses 53
\quad Using the \texttt{AND} Operator 53
\quad Using the \texttt{OR} Operator 54
\quad Understanding Order of Evaluation 55
Using the \texttt{IN} Operator 57
Using the \texttt{NOT} Operator 58
Summary 59

8: Using Wildcard Filtering 61
Using the \texttt{LIKE} Operator 61
\quad The Percent Sign (\%) Wildcard 62
\quad The Underscore (\_) Wildcard 64
Tips for Using Wildcards 65
Summary 65

9: Searching Using Regular Expressions 67
Understanding Regular Expressions 67
Using Regular Expressions 68
Basic Character Matching  68
Performing or Matches  70
Matching One of Several Characters  71
Matching Ranges  72
Matching Special Characters  73
Matching Character Classes  75
Matching Multiple Instances  75
Anchors  77
Summary  79

10:  Creating Calculated Fields  81
Understanding Calculated Fields  81
Concatenating Fields  82
   Using Aliases  84
Performing Mathematical Calculations  85
Summary  87

11:  Using Data Manipulation Functions  89
Understanding Functions  89
Using Functions  90
   Text Manipulation Functions  90
   Date and Time Manipulation Functions  92
   Numeric Manipulation Functions  96
Summary  96

12:  Summarizing Data  97
Using Aggregate Functions  97
   The AVG() Function  98
   The COUNT() Function  99
   The MAX() Function  100
   The MIN() Function  101
   The SUM() Function  102
Aggregates on Distinct Values  103
Combining Aggregate Functions  104
Summary  105

13:  Grouping Data  107
Understanding Data Grouping  107
Creating Groups  108
Filtering Groups  109
18: Full-Text Searching 147
   Understanding Full-Text Searching 147
   Using Full-Text Searching 148
      Enabling Full-Text Searching Support 148
      Performing Full-Text Searches 149
      Using Query Expansion 152
      Boolean Text Searches 154
   Full-Text Search Usage Notes 158
   Summary 159

19: Inserting Data 161
   Understanding Data Insertion 161
   Inserting Complete Rows 161
   Inserting Multiple Rows 165
   Inserting Retrieved Data 166
   Summary 168

20: Updating and Deleting Data 169
   Updating Data 169
   Deleting Data 171
   Guidelines for Updating and Deleting Data 172
   Summary 173

21: Creating and Manipulating Tables 175
   Creating Tables 175
      Basic Table Creation 176
      Working with NULL Values 177
      Primary Keys Revisited 179
      Using AUTO_INCREMENT 180
      Specifying Default Values 181
   Engine Types 182
   Updating Tables 183
   Deleting Tables 185
   Renaming Tables 185
   Summary 186
22: Using Views 187
Understanding Views 187
  Why Use Views 188
  View Rules and Restrictions 188
Using Views 189
  Using Views to Simplify Complex Joins 189
  Using Views to Reformat Retrieved Data 191
  Using Views to Filter Unwanted Data 192
  Using Views with Calculated Fields 193
Updating Views 194
Summary 195

23: Working with Stored Procedures 197
Understanding Stored Procedures 197
  Why Use Stored Procedures 198
Using Stored Procedures 199
  Executing Stored Procedures 199
  Creating Stored Procedures 200
  Dropping Stored Procedures 201
  Working with Parameters 202
  Building Intelligent Stored Procedures 205
  Inspecting Stored Procedures 208
Summary 208

24: Using Cursors 209
Understanding Cursors 209
  Working with Cursors 209
    Creating Cursors 210
    Opening and Closing Cursors 210
    Using Cursor Data 212
Summary 216

25: Using Triggers 217
Understanding Triggers 217
  Creating Triggers 218
  Dropping Triggers 219
Using Triggers 219
  INSERT Triggers 219
  DELETE Triggers 221
  UPDATE Triggers 223
  More on Triggers 223
Summary 224

26: Managing Transaction Processing 225
  Understanding Transaction Processing 225
  Controlling Transactions 227
    Using ROLLBACK 227
    Using COMMIT 228
    Using Savepoints 229
    Changing the Default Commit Behavior 230
Summary 230

27: Globalization and Localization 231
  Understanding Character Sets and Collation Sequences 231
  Working with Character Set and Collation Sequences 232
Summary 234

28: Managing Security 235
  Understanding Access Control 235
  Managing Users 236
    Creating User Accounts 237
    Deleting User Accounts 238
    Setting Access Rights 238
    Changing Passwords 241
Summary 242

29: Database Maintenance 243
  Backing Up Data 243
  Performing Database Maintenance 243
  Diagnosing Startup Problems 245
  Review Log Files 245
Summary 246

30: Improving Performance 247
  Improving Performance 247
Summary 249
A: Getting Started with MariaDB  251
   What You Need  251
   Obtaining the Software  252
   Installing the Software  252
   Preparing to Try It Yourself  253

B: The Example Tables  255
   Understanding the Sample Tables  255
      Table Descriptions  256
   Creating the Sample Tables  259
      Using mysql  260
      Using MySQL Workbench  261

C: MariaDB Datatypes  263
   String Datatypes  263
   Numeric Datatypes  265
   Date and Time Datatypes  266
   Binary Datatypes  266

D: MariaDB Reserved Words  269

Index  275
Foreword

As the creator of MariaDB (and MySQL), I am thrilled to see the first MariaDB book in print. I am equally thrilled that Ben Forta wrote it. Ben has a gift for presenting complex topics (and really understanding SQL can be complex) in an easy-to-understand way. MariaDB Crash Course is an easy read and goes from explaining the basics to the very complex (including joins, regular expressions, and triggers) simply and without painful effort. I recommend this book to anyone new to SQL who wants to quickly learn how to get the best out of MariaDB.

Michael “Monty” Widenius
Creator of MariaDB and MySQL
Acknowledgments

I’d like to thank the folks at Addison-Wesley for once again granting me the flexibility and freedom to build this book as I saw fit. Special thanks to Mark Taber for helping turn this one around in record time, and for his guidance into what this series is evolving into.

Thanks to project editor Elaine Wiley for keeping the project moving and me on schedule, no easy task.

Thanks to Monty Widenius, (creator of MariaDB and MySQL), Daniel Bartholomew, and Colin Charles for their thorough technical review and feedback.

And finally, this book was written in response to an unsolicited request by Monty Widenius. Monty is the driving force behind some of the most successful database projects in history, and yet he still took the time to review the manuscript, provide feedback, and write a much-appreciated foreword and recommendation. Thank you for your time and support, Monty. I hope this title lives up to your expectations.
About the Author

Ben Forta is Adobe Systems’ Director of Developer Relations and has more than 20 years experience in the computer industry in product development, support, training, and product marketing. Ben is the author of the best-selling Sams Teach Yourself SQL in 10 Minutes (now in its third edition, and translated into more than a dozen languages), spinoff titles on MySQL and SQL Server T-SQL, ColdFusion Web Application Construction Kit and Advanced ColdFusion Application Development (both published by Adobe Press), Sams Teach Yourself Regular Expressions in 10 Minutes, as well as books on Flash, Java, Windows, and other subjects. He has extensive experience in database design and development, has implemented databases for several highly successful commercial software programs and Web sites, and is a frequent lecturer and columnist on Internet and database technologies. Ben lives in Oak Park, Michigan, with his wife, Marcy, and their seven children. Ben welcomes your e-mail at ben@forta.com and invites you to visit his Web site at http://forta.com/.
MariaDB is an offshoot of MySQL, one of the most popular database management systems in the world. From small development projects to some of the best-known and most prestigious sites on the Web, MySQL has proven itself to be a solid, reliable, fast, and trusted solution to all sorts of data storage needs.

In 2008, MySQL was acquired by Sun Microsystems, which was in turn acquired by Oracle Corporation in 2010. While the initial acquisition by Sun was hailed by many in the MySQL community as exactly what the project needed, that sentiment did not last, and the subsequent acquisition by Oracle was unfortunately met with far lower expectations. Many of MySQL’s developers left Sun and Oracle to work on new projects. Among them was Michael “Monty” Widenius, creator of MySQL and one of the project’s longtime technical leads.

Monty and his team created a fork (offshoot) of the MySQL codebase and named his new DBMS MariaDB. The stated goals for the new MariaDB DBMS include

- Create a DBMS that is so compatible with MySQL that it could be used as a drop-in replacement (you could uninstall MySQL, install MariaDB, and your programs should continue to run as is). This is accomplished by building MariaDB on the MySQL codebase.
- Improve the source code to make MariaDB far more reliable and stable.
- Add features (and community contributions) at a faster rate.
- Develop a new underlying database engine (don’t worry if that sounds obscure for now) named Aria to improve performance and reliability.

What Is MariaDB Crash Course?

This book is based on my best-selling *Sams Teach Yourself SQL in 10 Minutes*. That book has become one of the most-used SQL tutorials in the world, with an emphasis on teaching what you really need to know—methodically, systematically, and simply. But as popular and as successful as that book is, it does have some limitations:
In covering all the major DBMSs, coverage of DBMS-specific features and functionality had to be kept to a minimum.

To simplify the SQL taught, the lowest common denominator had to be found—SQL statements that would (as much as possible) work with all major DBMSs. This requirement necessitated that better DBMS-specific solutions not be covered.

Although basic SQL tends to be rather portable between DBMSs, more advanced SQL most definitely is not. As such, that book could not cover advanced topics, such as triggers, cursors, stored procedures, access control, transactions, and more, in any real detail.

And that is where this book comes in. MariaDB Crash Course builds on the proven tutorials and structure of Sams Teach Yourself SQL in Ten Minutes, without getting bogged down with anything but MariaDB. Starting with simple data retrieval and working on to more complex topics, including the use of joins, subqueries, regular expression and full text-based searches, stored procedures, cursors, triggers, table constraints, and much more. You learn what you need to know methodically, systematically, and simply—in highly focused chapters designed to make you immediately and effortlessly productive.

Who Is This Book For?

This book is for you if

- You are new to SQL.
- You are just getting started with MariaDB and want to hit the ground running.
- You want to quickly learn how to get the most out of MariaDB.
- You want to learn how to use MariaDB in your own application development.
- You want to be productive quickly and easily using MariaDB without having to call someone for help.

It is worth noting that this book is not intended for all readers. If you are an experienced SQL user, you may find the content in this book too elementary. Similarly, if you have existing MySQL experience, you’ll likely find this book to be less useful (as noted, MariaDB is based on MySQL). If you own my MySQL Crash Course, I do not recommend that you buy this book, as much of
the content is similar, and your existing MySQL knowledge will easily transfer as is to MariaDB.

But, if the preceding list describes you and your needs relative to MariaDB, you’ll find this MariaDB Crash Course to be the fastest and easiest way to get up to speed with MariaDB.

This book is also useful if you are new to MySQL, as most of the content also applies to that DBMS. For you, this book has an extra benefit in that it helps demonstrate some reasons to consider switching to MariaDB.

Companion Web Site

This book has a companion Web site online at http://forta.com/books/0321799941/. Visit the site to access

- Table creation and population scripts used to create the example tables used throughout this book
- The online support forum
- Online errata (should one be required)
- Other books that may be of interest to you

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

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

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

New Term
Provides clear definitions of new, essential terms.

Input
The Input icon identifies code that you can type in yourself. It usually appears next to a listing.

Output
The Output icon highlights the output produced by running MariaDB code. It usually appears after a listing.

Analysis
The Analysis icon alerts you to the author’s line-by-line analysis of input or output.
Working with MariaDB

In this chapter, you learn how to connect and log in to MariaDB, how to issue MariaDB SQL statements, and how to obtain information about databases and tables.

Making the Connection

Note

Example Tables Required From this point on, all chapters will use the example databases and tables. If you have yet to install these, see Appendix B, “The Example Tables,” before proceeding.

Now that you have a MariaDB DBMS and client software to use with it, it would be worthwhile to briefly discuss connecting to the database.

MariaDB, like all client-server DBMSs, requires that you log in to the DBMS before being able to issue commands. Login names might not be the same as your network login name (assuming that you are using a network); MariaDB maintains its own list of users internally and associates rights with each.

When you first installed MariaDB, you may have been prompted for an administrative login (usually named root) and a password (if you weren’t, then the root user account was created with no password). If you are using your own local server and are simply experimenting with MariaDB, using this login is fine. In the real world, however, the administrative login is closely protected (as access to it grants full rights to create tables, drop entire databases, change logins and passwords, and more).

To connect to MariaDB you need the following pieces of information:

- The hostname (the name of the computer)—this is localhost if connecting to a local MariaDB server
- The port (if a port other than the default 3306 is used)
- A valid user name
- The user password (if required)
As explained in Chapter 2, “Introducing MariaDB,” all this information can be passed to the `mysql` command line utility, or entered into the server connection screen in MySQL Workbench.

**Note**

**Using Other Clients** If you are using a client other than the ones mentioned here, you still need to provide this information to connect to MariaDB.

After you are connected, you have access to whatever databases and tables your login name has access to. (Logins, access control, and security are revisited in Chapter 28, “Managing Security.”)

### Selecting a Database

When you first connect to MariaDB, you do not have any databases open for use. Before you can perform any database operations, you need to select a database. To do this you use the `USE` keyword.

#### New Term

**Keyword** A reserved word that is part of the MariaDB SQL language. Never name a table or column using a keyword. Appendix D, “MariaDB Reserved Words,” lists the MariaDB keywords.

For example, to use the `crashcourse` database you would enter the following:

**Input**

```
USE crashcourse;
```

**Output**

```
Database changed
```

**Analysis**

The `USE` statement does not return any results. Depending on the client used, some form of notification might be displayed. For example, the `Database changed` message shown here is displayed by the `mysql` command line utility upon successful database selection.

#### Tip

**Preselecting a Database** If you are using the `mysql` command line tool, you can pre-select a database by typing its name after `mysql` when running the tool.
Remember, you must always USE a database before you can access any data in it.

**Learning About Databases and Tables**

But what if you don’t know the names of the available databases? And for that matter, how are clients like MySQL Workbench able to display a list of available databases?

Information about databases, tables, columns, users, privileges, and more is stored within databases and tables themselves (yes, MariaDB uses MariaDB to store this information). But these internal tables are generally not accessed directly. Instead, the MariaDB SHOW command can be used to display this information (information that MariaDB then extracts from those internal tables). Look at the following example:

▼ **Input**

```
SHOW DATABASES;
```

▼ **Output**

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| crashcourse        |
| mysql              |
| forta              |
| coldfusion         |
| flex               |
| test               |
+--------------------+
```

▼ **Analysis**

`SHOW DATABASES;` returns a list of available databases. Included in this list might be databases used by MariaDB internally (such as `mysql` and `information_schema` in this example). Of course, your own list of databases might not look like those shown here.

To obtain a list of tables within a database, use `SHOW TABLES;`, as seen here:

▼ **Input**

```
SHOW TABLES;
```
### Output

```
+-----------------------+
| Tables_in_crashcourse |
+-----------------------+
| customers             |
| orderitems            |
| orders                |
| products              |
| productnotes          |
| vendors               |
+-----------------------+
```

### Analysis

SHOW TABLES; returns a list of available tables in the currently selected database.

To show a table’s columns, you can use DESCRIBE:

### Input

```sql
DESCRIBE customers;
```

### Output

```
+--------------+-----------+------+-----+---------+----------------+
| Field        | Type      | Null | Key | Default | Extra          |
|--------------+-----------+------+-----+---------+----------------+
| cust_id      | int(11)   | NO   | PRI | NULL    | auto_increment |
| cust_name    | char(50)  | NO   |     |         |                |
| cust_address | char(50)  | YES  |     | NULL    |                |
| cust_city    | char(50)  | YES  |     | NULL    |                |
| cust_state   | char(5)   | YES  |     | NULL    |                |
| cust_zip     | char(10)  | YES  |     | NULL    |                |
| cust_country | char(50)  | YES  |     | NULL    |                |
| cust_contact | char(50)  | YES  |     | NULL    |                |
| cust_email   | char(255) | YES  |     | NULL    |                |
+--------------+-----------+------+-----+---------+----------------+
```

### Analysis

DESCRIBE requires that a table name be specified (`customers` in this example), and returns a row for each field containing the field name, its datatype, whether NULL is allowed, key information, default value, and extra information (such as `auto_increment` for field `cust_id`).
Learning About Databases and Tables

Note

What Is Auto Increment? Some table columns need unique values. For example, order numbers, employee IDs, or (as in the example just seen) customer IDs. Rather than have to assign unique values manually each time a row is added (and having to keep track of what value was last used), MariaDB can automatically assign the next available number for you each time a row is added to a table. This functionality is known as auto increment. If it is needed, it must be part of the table definition used when the table is created using the CREATE statement. We look at CREATE in Chapter 21, “Creating and Manipulating Tables.”

Tip

The SHOW COLUMNS FROM Statement DESCRIBE is actually a shortcut for SHOW COLUMNS FROM. In other words, the statement DESCRIBE customers; is functionally identical to the statement SHOW COLUMNS FROM customers;.

Other SHOW statements are supported too, including

- SHOW STATUS—Used to display extensive server status information
- SHOW CREATE DATABASE and SHOW CREATE TABLE—Used to display the MariaDB statements used to create specified databases or tables respectively
- SHOW GRANTS—Used to display security rights granted to users (all users or a specific user)
- SHOW ERRORS and SHOW WARNINGS—Used to display server error or warning messages

It is worthwhile to note that client applications use these same MariaDB SQL commands as you’ve seen here. Applications that display interactive lists of databases and tables, that allow for the interactive creation and editing of tables, that facilitate data entry and editing, or that allow for user account and rights management, and more, all accomplish what they do using the same MariaDB SQL commands that you can execute directly yourself.

Tip

Learning More About SHOW In the mysql command line utility, execute command HELP SHOW; to display a list of allowed SHOW statements.

Note

Want Even More Information? MariaDB supports the use of INFORMATION_SCHEMA to obtain and filter even more schema details. Coverage of INFORMATION_SCHEMA is beyond the scope of this book. But, if you should need it, know that it’s there for you.
Summary

In this chapter, you learned how to connect and log in to MariaDB; how to select databases using `USE`; and how to introspect MariaDB databases, tables, and internals using `SHOW` and `DESCRIBE`. Armed with this knowledge, you can now dig into the all-important `SELECT` statement.
Symbols

* (asterisk), 30
\ (backslash), 74
/* */ comment syntax, 36
% (percent sign) wildcard, 62-63
# (pound sign), 36
; (semicolon), 28
-- (two hyphens), 35
_ (underscore) wildcard, 64

A

Abs() function, 96
access control, 235-236
access rights, 238-241
accounts. See user accounts
advantages of MySQL, 13-14
Against() function, 149-152
aggregate functions
  ALL argument, 103
  AVG(), 98-99
  combining, 104-105
  COUNT(), 99-100
  defined, 97
  DISTINCT argument, 103-104
  explained, 97
  joins and, 139-140
  MAX(), 100-101
  MIN(), 101-102
  naming aliases, 105
  SUM(), 102-103

aliases, 84-85, 133-134
ALL argument, 103
alphabetical sort order, 40-43
ALTER TABLE statement, 183-185
ANALYZE TABLE statement, 244
anchor metacharacters, 77
anchors, 77-79
AND keyword, 50
AND operator, 53-54
application filtering, 46
AS keyword, 84-85
asterisk (*), 30
auto increment, 25
AUTO_INCREMENT, 180-181
AVG() function, 98-99

B

backing up data, 243
backslash (\), 74
BACKUP TABLE statement, 243
BETWEEN operator (WHERE clause), 49
BINARY datatype, 266-267
BIT datatype, 265
boolean text searches, 154-158

calculated fields
  aliases, 84-85
  concatenating fields, 82-83
  explained, 81-82
mathematical calculations, 85-86
subqueries as, 119-121
views, 193-194
calculated values, totaling, 102
calling stored procedures, 199
Cartesian products, 127
case sensitivity, 28, 42
changing passwords, 241-242
character classes, matching, 75
character matching
anchors, 77-79
basic character matching, 68-70
character classes, 75
multiple instances, 75-77
one of several characters, 71-72
OR matches, 70
ranges, 72-73
special characters, 73-74
character sets, 232-234
checking
for nonmatches, 48-49
for NULL value, 50-51
for range of values, 49-50
against single value, 47
CHECK TABLE statement, 244
clauses. See also specific clauses
definition of, 38
positioning, 46, 51
SELECT clause ordering, 113-114
client-based results formatting, 82
client-server software, 14-15
CLOSE statement, 211
closing cursors, 211
collation sequences, 232-234
columns. See also fields
aliases, 84-85
derived, 85
explained, 7-8
fully qualified names, 126
GROUP BY clause, 109
multiple, sorting query results by, 39-40
NULL, 8, 177-178
omitting, 164
padded spaces
RTrim() function, 83-84
primary keys, 9-10
retrieving
all columns, 30
individual columns, 27-28
multiple columns, 29
unknown columns, 31
subquery result restrictions, 118
updating multiple, 170
values, deleting, 171
viewing, 24
combined queries
creating, 141-144
duplicate rows and, 144-145
explained, 141
sorting results, 145-146
combining
aggregate functions, 104-105
WHERE clauses
AND operator, 53-54
OR operator, 54-55
order of evaluation, 55-56
comments, 35-36
commits, 227
default commit behavior, 230
explicit commits, 228-229
implicit commits, 228
COMMIT statement, 228-229
compatibility with MySQL, 15
compound queries. See combined queries
concatenating fields, 82-83
Concat() function, 82
conditional operators, 46
correlated subqueries, 120
Cos() function, 96
DATA

breaking correctly (columns), 7

DELETING

guidelines, 172-173

TRUNCATE TABLE

statement, 172

FILTERING. See data filtering

GROUPING

explained, 107

filtering groups, 109-111

GROUP BY clause, 108-109

grouping and sorting, 112-113

nested groups, 108

UPDATING, 172-173

databases. See also tables

explained, 5-6

MAINTENANCE

backing up data, 243
diagnosing startup problems, 245
performing, 243-244

reviewing log files, 245-246

SCHEMAS, 7

SELECTING, 22

VIEWING AVAILABLE DATABASES, 23

database servers, 14

data filtering

GROUPS, 109-111

by subqueries, 115-118

WHERE clause, 45-46

checking against single value, 47

checking for nonmatches, 48-49

checking for NULL value, 50-51

checking for range of values, 49-50

combining clauses, 53-56

conditional operators, 46

IN operator, 57-58

NOT operator, 58-59

wildcard filtering

LIKE operator, 61

% (percent sign) wildcard, 62-63

tips, 65

_ (underscore) wildcard, 64

with views, 192

datatypes, 8

binary, 266-267
currency, 266
date and time, 266
numeric, 265-266
string, 263-264

usefulness of, 263

date and time datatypes, 266
date and time functions, 90-95

DATE data type, 266

DATETIME data type, 266

DBMS (Database Management System), 6, 61

DECIMAL data type, 265

DECLARE statement, 210, 213-214

default commit behavior, 230

default values, 181-182

DELAYED keyword, 248

DELETE statement, 171-172

FROM clause, 172
guidelines, 172-173
triggers, 221-222
WHERE clause, 171

deleting

column values, 171
data

guidelines, 172-173
table data, 171-172
TRUNCATE TABLE statement, 172

tables, 185
user accounts, 238

derived columns. See aliases

DESC keyword, 40-43

DESCRIBE statement, 24-25

diagnosing startup problems, 245

dictionary sort order, 42

DISTINCT keyword, 31-32, 103-104

downloading

MariaDB, 252
MySQL Workbench, 252

DROP PROCEDURE statement, 201

DROP TABLE statement, 185

DROP TRIGGER statement, 219

DROP USER statement, 238

dropping

stored procedures, 201
triggers, 219

empty strings, 179

enabling full-text searching, 148-149

encodings, 231

engine types, 182-183

equijoins. See inner joins

escaping, 74

evaluation, order of, 55-56

EXECUTE statement, 199

executing

scripts, 17
stored procedures, 199

Exp() function, 96

EXPLAIN statement, 248

explicit commits, 228-229

expressions. See regular expressions

FETCH statement, 212, 215-216

fields, calculated. See also columns

aliases, 84-85
concatenating, 82-83
explained, 81-82
mathematical calculations, 85-86
subqueries as, 119-121
views, 193-194

files, log

flushing, 246
reviewing, 245-246

filtering data

application filtering, 46
by subqueries, 115-118
groups, 109-111
WHERE clause, 45-46

checking against single value, 47
checking for nonmatches, 48-49
checking for NULL value, 50-51
checking for range of values, 49-50

diagnosing startup problems, 245

dictionary sort order, 42

downloading

MariaDB, 252
MySQL Workbench, 252

DROP PROCEDURE statement, 201

DROP TABLE statement, 185

DROP TRIGGER statement, 219

DROP USER statement, 238

dropping

stored procedures, 201
triggers, 219
combining clauses, 53-56
conditional operators, 46
wildcard filtering
LIKE operator, 61
tips, 65
% (percent sign) wildcard, 62-63
_ (underscore) wildcard, 64
with views, 192

fixed length strings, 264
FLOAT datatype, 265
flushing log files, 246
FLUSH LOGS statement, 246
FLUSH TABLES statement, 243
foreign keys, 124
formatting
retrieved data with views, 191
server-based compared to client-based, 82
statements, 177
subqueries, 117
FROM keyword, 27
FULLTEXT clause, 148-149
full-text searching
boolean text searches, 154-158
enabling, 148-149
explained, 147-148
performing searches, 149-152
query expansion, 152-154
support for, 147
tips and guidelines, 158
fully qualified table names, 34-35
functions
Against(), 149-152
aggregate functions, 97
ALL argument, 103
AVG(), 98-99
combining, 104-105
COUNT(), 99-100
DISTINCT argument, 103-104
joins and, 139-140
MAX(), 100-101
MIN(), 101-102
SUM(), 102-103
Concat(), 82
date and time functions, 92-95
defined, 89
LTrim(), 84
Match(), 149-152
numeric functions, 96
portability, 89
RTrim(), 83-84
system, 90
text functions, 90-92
Trim(), 84

G

globalization, 232-234
granting access rights, 239
GRANT statement, 239
GROUP BY clause, 108-109
grouping data
explained, 107
filtering groups, 109-111
GROUP BY clause, 108-109
grouping and sorting, 112-113
nested groups, 108
groups, filtering, 109-111

H

HAVING clause, 109
help option (mysqld), 245
HELP SHOW statement, 25
natural joins, 136-137
outer joins, 137-138
self joins, 134-136
simplifying with views, 189-190

keys. See foreign keys; primary keys
keywords, 22, 269-274. See also specific keywords
KILL statement, 248

Left() function, 91
Length() function, 91
less than operator (WHERE clause), 156
LIKE keyword, 147
LIKE operator, 61
LIMIT clause, 32-34
Locate() function, 91
log-bin command line option, 245
log command line option, 245
log-error command line option, 245
log files
flushing, 246
reviewing, 245-246
logging in, 21-22
logical operators. See operators
login names, 21-22
log-slow-queries command line option, 246
Lower() function, 91
LTrim() function, 84, 91

maintenance (database)
backing up data, 243
diagnosing startup problems, 245
performing, 243-244
reviewing log files, 245-246
null values
checking for, 50-51
compared to empty strings, 179
COUNT() function, 100
null columns, 8
primary keys, 180
table columns, 177-178
numeric datatypes, 265-266
numeric functions, 90, 96
numeric values, storing, 265

OPEN CURSOR statement, 210-211
opening cursors, 210-211
operators
AND, 53-54
boolean operators, 156
conditional operators, 46
definition of, 53
IN, 57-58
LIKE, 61
mathematical, 86
NOT, 58-59
OR, 54-55
OPTIMIZE TABLE statement, 244
optimizing performance, 24-29
OR operator, 54-55, 70
ORDER BY clause, 38-43, 54, 112-113
positioning, 43
sorting by multiple columns, 40
orderitems table, 258
order of evaluation, 55-56
orders table, 257-258
outer joins, 137-138
overriding AUTO_INCREMENT, 181
overwriting tables, 177
parameters for stored procedures, 202-205
passwords, 241-242
percent sign (%) wildcard, 62-63

performance
   deleting data, 172
   improving, 247-249
   subqueries, 119
   views, 188

phone numbers, 265
Pi() function, 96
placeholders. See savepoints
populate.sql script, 261-262
portability, 89
pound sign (#), 36
predicates, 62
primary keys, 124, 179-180
   customer table, 257
   explained, 9-10
   importance, 9
   orders table, 258
   products, 259
   products table, 257
   vendors table, 256
privileges. See access rights
procedures, stored. See stored procedures
processing transactions. See transaction processing
productnotes table, 259
productpricing() stored procedure, 200-201
products table, 256-257
proximity searching, 159

queries
   calculated fields
      aliases, 84-85
      concatenating fields, 82-83
   explained, 81-82
   mathematical calculations, 85-86
   data formatting, 30
   defined, 115
   sorting results, 37-38
      ascending/descending order, 40-43
      by multiple columns, 39-40
      case sensitivity, 42
      nonselected columns and, 39
   views, 187
query expansion, 152-154
quotes
   numeric values, 266
   string values, 264

Rand() function, 96
range of values, checking for, 49-50
ranges, matching, 72-73
REAL datatype, 265
records, 8
referential integrity, 125
REGEXP keyword, 69-70
regular expressions, 67-68
   anchors, 77-79
   basic character matching, 68-70
   matching
      character classes, 75
      multiple instances, 75-77
      one of several characters, 71-72
      ranges, 72-73
      special characters, 73-74
   OR matches, 70
relational tables, 123-124
RENAME TABLE statement, 185
renaming tables, 185
REPAIR TABLE statement, 244
repetition metacharacters, 76
replacing tables, 177
reserved words, 22, 269-274
RESTORE TABLE statement, 243
retrieved data, inserting, 166-168
retrieving
columns
   all columns, 30
   individual columns, 27-28
   multiple columns, 29
   unknown columns, 31
   rows, 31-32
reusable views, creating, 190
reviewing log files, 245-246
REVOKE statement, 239-240
revoking access rights, 239-240
Right() function, 91
RIGHT keyword (outer joins), 138
rights. See access rights
rollbacks
   COMMIT statement, 228-229
   ROLLBACK statement, 227-228
ROLLBACK statement, 227-228
rows
cursors, 209
explained, 8
inserting
   multiple rows, 165-166
   single rows, 161-164
NULL, 8
retrieving, 31-32
RTrim() function, 83-84, 91
S
safe-mode option (mysqld), 245
Sams Teach Yourself Regular Expressions
   in 10 Minutes, 68
savepoints, 227-230
SAVEPOINT statement, 229
scalability, 124
scripts
   create.sql, 262
   executing, 17
   populate.sql, 261-262
search criteria. See filtering data
searching
   full-text searching
      boolean text searches, 154-158
      enabling, 148-149
      explained, 147-148
      performing searches, 149-152
      query expansion, 152-154
      support for, 147
      tips and guidelines, 158
   proximity searching, 159
   with regular expressions. See regular expressions
search pattern, 61
security
   access control, 235-236
   user accounts
      access rights, 238-241
      creating, 237-238
      deleting, 238
      obtaining list of, 236-237
      passwords, 241-242
SELECT statement, 27
   AS keyword, 84-85
   clause ordering, 113-114
   concatenating fields, 83
   DISTINCT keyword, 31-32
   FROM keyword, 27
   fully qualified table names, 34-35
   GROUP BY clause, 108-109
   IN operator, 57-58
   INSERT SELECT, 166-168
   IS NULL clause, 50
   LIMIT clause, 32-34
limiting results, 32-34
NOT operator, 58-59
ORDER BY clause, 38-43, 112-113
retrieving, 31-32
retrieving all columns, 30
retrieving individual columns, 27-28
retrieving multiple columns, 29
retrieving unknown columns, 31
SELECT INTO OUTFILE, 243
WHERE clause, 45-46
    checking against single value, 47
    checking for nonmatches, 48-49
    checking for NULL value, 50-51
    checking for range of values, 49-50
    combining clauses, 53-56
    conditional operators, 46

selecting databases, 22
self joins, 134-136
semscolons (;), 28
server-based results formatting, 82
servers, database servers, 14
SET PASSWORD statement, 241-242
SET statement, 170
SHOW CHARACTER SET statement, 232
SHOW COLLATION statement, 232
SHOW COLUMNS FROM statement, 25
SHOW CREATE DATABASE statement, 25
SHOW CREATE TABLE statement, 25
SHOW DATABASES statement, 23
SHOW ERRORS statement, 25
SHOW GRANTS statement, 25, 238
SHOW PROCESSLIST statement, 248
SHOW STATUS statement, 25
SHOW TABLES statement, 23-24
showing
    columns, 24
databases, 23
tables, 23-24
Sin() function, 96

single value, checking against, 47
SMALLINT datatype, 265-266
sorting query results, 37-38, 112-113
    ascending/descending order, 40-43
case sensitivity, 42
by multiple columns, 39-40
nonselected columns and, 39
Soundex() function, 91
spaces, removing, 83-84
special characters, matching, 73-74
SQL statements. See specific statements
Sqrt() function, 96
standard deviation aggregate functions, 98
startup problems, diagnosing, 245
statements. See specific statements
stopwords, 158
stored procedures
    advantages of, 198
    building intelligent stored procedures, 205-207
creating, 200-201
disadvantages of, 198
dropping, 201
executing, 199
explained, 197-198
inspecting, 208
parameters, 202-205
productpricing(), 200-201

storing
    date and time values, 266
    numeric values, 265
    strings, 263

string datatypes, 263-264
strings. See text functions
    empty, 179
    fixed length, 264
    quotes, 264
    variable-length, 264
    wildcard searching and, 62
subqueries
   as calculated fields, 119-121
   combining, 117
   correlated subqueries, 120
   defined, 115
   explained, 115
   filtering by, 115-118
   formatting, 117
   performance and, 119
   UPDATE statement, 170
   WHERE clause, 118
SubString() function, 91
SUM() functions, 98, 102-103

tables
   aliases, 133-134
   AUTO_INCREMENT, 180-181
   calculated fields
      aliases, 84-85
      concatenating fields, 82-83
      explained, 81-82
      mathematical calculations, 85-86
   Cartesian products, 127
   columns
      aliases, 84-85
      explained, 7-8
      NULL, 8
      primary keys, 9-10
      retrieving all columns, 30
      retrieving individual columns, 27-28
      retrieving multiple columns, 29
      retrieving unknown columns, 31
   creating, 259
      CREATE TABLE statement, 176-177
      explained, 175
   with mysql, 260-261
   with MySQL Workbench, 261-262
   customers, 257
   default values, 181-182
   deleting, 185
   deleting data from, 171-172
   explained, 6-7
   foreign keys, 124
   fully qualified table names, 34-35
   functions of, 255
   inserting retrieved data, 166-168
   joins
      advantages of, 125
      with aggregate functions, 139-140
      creating, 125-129
      explained, 123
      inner joins, 129-130
      join conditions, 140
      joining multiple tables, 130-132
      natural joins, 136-137
      outer joins, 137-138
      self joins, 134-136
   naming, 7
   NULL values, 50, 177-178
   orderitems, 258
   orders, 257-258
   overwriting, 177
   performance considerations, 131
   primary keys, 124, 179-180
   productnotes, 259
   products, 256-257
   relational tables, 123-124
   renaming, 185
   replacing, 177
   rows
      explained, 8
      inserting multiple rows, 165-166
      inserting single rows, 161-164
NULL, 8
  retrieving, 31-32
  updating, 169-171, 183-185
  vendors, 256
  viewing list of, 23
  virtual. See views
Tan() function, 96
terminating statements, 28
text functions, 90-92
text searches. See full-text searching
time functions, 92-95
TINYINT datatype, 266
transaction processing
  COMMIT statement, 228-229
  default commit behavior, 230
  explained, 225-226
  explicit commits, 228-229
  implicit commits, 228
  managing, 227
  ROLLBACK statement, 227-228
  SAVEPOINT statement, 229
  savepoints, 229-230
  support for, 225
  terminology, 227
transactions, 227
triggers
  creating, 218
  DELETE, 221-222
  dropping, 219
  explained, 217
  INSERT, 219-221
  tips and guidelines, 223-224
  UPDATE, 223
Trim() function, 84
trimming padded spaces, 83-84
TRUNCATE TABLE statement, 172

U
underscore (_), wildcard, 64
UNION keyword, 142-145
unions. See combined queries
UPDATE statement, 169-171
  guidelines, 172-173
  subqueries, 170
UPDATE triggers, 223
updating data, 172-173
  tables, 169-171, 183-185
  views, 194
Upper() function, 91
USE statement, 22
user accounts
  access rights, 238-241
  creating, 237-238
  deleting, 238
  obtaining list of, 236-237
  passwords, 241-242
utilities. See specific utilities

V
VARBINARY datatype, 264, 267
variable-length strings, 264
vendors table, 256
verbose option (mysqld), 245
version option (mysqld), 245
viewing
  available databases, 23
  available tables, 23
  columns, 24
views
  advantages of, 188
  calculated fields, 193-194
  creating, 189
  explained, 187
  filtering data, 192
performance concerns, 188
reformatting retrieved data, 191
reusable views, 190
rules and restrictions, 188-189
simplifying joins with, 189-190
updating, 194

virtual tables. See views

W-X-Y-Z

WHERE clause, 45-46, 109-110, 145
checking against single value, 47
checking for nonmatches, 48-49
checking for NULL value, 50-51
checking for range of values, 49-50
combined queries, 141
combining clauses
    AND operator, 53-54
    OR operator, 54-55
    order of evaluation, 55-56
conditional operators, 46
DELETE statements, 171
IN operator, 57-58
joins, 129
NOT operator, 58-59
parentheses and, 56
subqueries, 118
UPDATE statements, 169-170
wildcards, 61

white space in SQL statements, 29
whitespace metacharacters, 74

Widenius, Michael, 14

wildcard filtering
    LIKE operator, 61
    % (percent sign) wildcard, 62-63
tips, 65
    _ (underscore) wildcard, 64