

MYSQL

IN A NUTSHELL

A Desktop Quick Reference

O'REILLY®

Russell J.T. Dyer

MYSQL IN A NUTSHELL



MySQL, and the APIs used for interacting with it, are so rich and complex that it's hard to remember all the possible nuances. *MySQL in a Nutshell* provides the syntax with explanations that are easy to understand, along with examples to help you obtain a comprehensive view and a layout that makes retrieving information quick and simple.

After introductory material on installing and formulating queries to MySQL, the book presents a reference listing of MySQL's features. Contents include:

- SQL statements, with the clauses of the more complicated statements broken into groups of related functionality for easier reading
- Functions provided by MySQL
- Arguments and configuration options for the *mysqld* server
- Arguments for the terminal client and a number of MySQL-related utilities, both within and outside the core distribution
- API functions for Perl, PHP, and C

All function and API listings are provided with a wealth of examples based on realistic applications, to show how the features can benefit you and make for simpler, faster, more productive queries. The information in this book can make every MySQL user more productive—novice and veteran alike.

O'REILLY® www.oreilly.com

US \$39.95

CAN \$55.95

ISBN-10: 0-596-00789-2

ISBN-13: 978-0-596-00789-8



Includes
FREE 45-Day
Online Edition

MYSQL

IN A NUTSHELL

Other resources from O'Reilly

Related titles	Web Database Applications with PHP and MySQL	MySQL Cookbook
	Essential PHP Security	Learning SQL
	PHP Cookbook	SQL in a Nutshell
	Programming PHP	SQL Pocket Guide
	Managing and Using MySQL	Learning Perl

oreilly.com *oreilly.com* is more than a complete catalog of O'Reilly books. You'll also find links to news, events, articles, weblogs, sample chapters, and code examples.



oreillynet.com is the essential portal for developers interested in open and emerging technologies, including new platforms, programming languages, and operating systems.

Conferences O'Reilly brings diverse innovators together to nurture the ideas that spark revolutionary industries. We specialize in documenting the latest tools and systems, translating the innovator's knowledge into useful skills for those in the trenches. Visit *conferences.oreilly.com* for our upcoming events.



Safari Bookshelf (*safari.oreilly.com*) is the premier online reference library for programmers and IT professionals. Conduct searches across more than 1,000 books. Subscribers can zero in on answers to time-critical questions in a matter of seconds. Read the books on your Bookshelf from cover to cover or simply flip to the page you need. Try it today with a free trial.

MYSQL

IN A NUTSHELL

Russell J.T. Dyer

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

MySQL in a Nutshell

by Russell J.T. Dyer

Copyright © 2005 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor: Andy Oram
Production Editor: Sarah Sherman
Cover Designer: Ellie Volckhausen
Interior Designer: David Futato

Printing History:

May 2005: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. The *In a Nutshell* series designations, *MySQL in a Nutshell*, the image of a pied kingfisher, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 0-596-00789-2

ISBN13: 978-0-596-00789-8

[M]

[05/07]

*To my friend Richard Stringer, for
encouraging me in literature, liberalism,
and writing, and for helping me to
become the person I was meant to be.*

—Russell J.T. Dyer

Table of Contents

Preface	xxi
1. Introduction to MySQL	1
The Value of MySQL	1
The MySQL Package	2
Licensing	3
Mailing Lists	3
Books and Other Publications	4
2. Installing MySQL	5
Choosing a Distribution	5
Unix Source Distributions	6
Unix Binary Distributions	8
Linux RPM Distributions	9
Macintosh OS X Distributions	10
Novell NetWare Distributions	11
Windows Distributions	12
Postinstallation	13
3. MySQL Basics	16
The mysql Client	16
Creating a Database and Tables	17
Show Me	19
Inserting Data	20
Selecting Data	21
Ordering, Limiting, and Grouping	22

Analyzing and Manipulating Data	23
Changing Data	25
Deleting Data	26
Searching Data	27
Importing Data in Bulk	28
Command-Line Interface	30
Conclusion	31
4. SQL Statements	32
Statements Grouped by Type	32
Statements and Clauses in Alphabetical Order	33
ALTER DATABASE	34
ALTER TABLE	34
ALTER VIEW	38
ANALYZE TABLE	39
BACKUP TABLE	39
CACHE INDEX	40
CHANGE MASTER TO	40
CHECK TABLE	41
CHECKSUM TABLE	42
COMMIT	42
CREATE DATABASE	43
CREATE INDEX	43
CREATE TABLE	44
CREATE VIEW	49
DELETE	50
DESCRIBE	51
DO	52
DROP DATABASE	52
DROP INDEX	52
DROP TABLE	53
DROP USER	53
DROP VIEW	54
EXPLAIN	54
FLUSH	54
GRANT	55
HANDLER	57
INSERT	59
JOIN	62
KILL	64
LOAD DATA FROM MASTER	64

LOAD DATA INFILE	65
LOAD INDEX INTO CACHE	66
LOAD TABLE...FROM MASTER	66
LOCK TABLES	67
OPTIMIZE TABLE	67
PURGE MASTER LOGS	68
RENAME TABLE	68
REPAIR TABLE	69
REPLACE	70
RESET	71
RESET MASTER	71
RESET SLAVE	71
RESTORE TABLE	72
REVOKE	72
ROLLBACK	73
ROLLBACK TO SAVEPOINT	73
SAVEPOINT	73
SELECT	74
SET	80
SET PASSWORD	81
SET SQL_LOG_BIN	81
SET TRANSACTION	81
SHOW BINLOG EVENTS	82
SHOW CHARACTER SET	82
SHOW COLLATION	83
SHOW COLUMNS	83
SHOW CREATE DATABASE	84
SHOW CREATE TABLE	84
SHOW CREATE VIEW	85
SHOW DATABASES	85
SHOW ENGINES	85
SHOW ERRORS	85
SHOW GRANTS	86
SHOW INDEX	86
SHOW INNODB STATUS	87
SHOW LOGS	87
SHOW MASTER LOGS	87
SHOW MASTER STATUS	88
SHOW PRIVILEGES	88
SHOW PROCESSLIST	88
SHOW SLAVE HOSTS	89

SHOW SLAVE STATUS	89
SHOW STATUS	90
SHOW TABLE STATUS	90
SHOW TABLES	91
SHOW VARIABLES	91
SHOW WARNINGS	92
START SLAVE	92
START TRANSACTION	92
STOP SLAVE	93
TRUNCATE TABLE	93
UNION	93
UNLOCK TABLES	94
USE	94
5. String Functions	95
String Functions Grouped by Type	95
String Functions in Alphabetical Order	96
AES_DECRYPT()	96
AES_ENCRYPT()	96
ASCII()	97
BIN()	97
BINARY	97
BIT_LENGTH()	98
CHAR()	98
CHAR_LENGTH()	98
CHARACTER_LENGTH()	99
COMPRESS()	99
CONCAT()	99
CONCAT_WS()	100
CONV()	100
DECODE()	100
DES_DECRYPT()	101
DES_ENCRYPT()	101
ELT()	101
ENCODE()	102
ENCRYPT()	102
EXPORT_SET()	102
FIELD()	103
FIND_IN_SET()	103
HEX()	103
INET_ATON()	104

INET_NTOA()	104
INSERT()	104
INSTR()	105
LCASE()	105
LEFT()	105
LENGTH()	106
LOAD_FILE()	106
LOCATE()	106
LOWER()	107
LPAD()	107
LTRIM()	107
MAKE_SET()	108
MATCH () AGAINST()	108
MD5()	108
MID()	109
OCT()	109
OCTET_LENGTH()	109
OLD_PASSWORD()	109
ORD()	110
PASSWORD()	110
POSITION()	110
QUOTE()	111
REPEAT()	111
REPLACE()	111
REVERSE()	111
RIGHT()	112
RPAD()	112
RTRIM()	112
SHA()	113
SHA1()	113
SOUNDEX()	113
SPACE()	113
STRCMP()	114
SUBSTRING()	114
SUBSTRING_INDEX()	114
TRIM()	115
UCASE()	115
UNCOMPRESS()	116
UNCOMPRESSED_LENGTH()	116
UNHEX()	116
UPPER()	116

6. Date and Time Functions	117
Date and Time Functions Grouped by Type	118
Determining the Date and Time	118
Extracting and Formatting the Date and Time	118
Calculating and Modifying the Date and Time	118
Date and Time Functions in Alphabetical Order	118
ADDDATE()	118
ADDTIME()	119
CONVERT_TZ()	119
CURDATE()	120
CURRENT_DATE()	120
CURRENT_TIME()	120
CURRENT_TIMESTAMP()	121
CURTIME()	121
DATE()	121
DATE_ADD()	122
DATE_FORMAT()	122
DATE_SUB()	124
DATEDIFF()	124
DAY()	124
DAYNAME()	125
DAYOFMONTH()	125
DAYOFWEEK()	125
DAYOFYEAR()	126
EXTRACT()	126
FROM_DAYS()	126
FROM_UNIXTIME()	127
GET_FORMAT()	127
HOUR()	128
LAST_DAY()	129
LOCALTIME()	129
LOCALTIMESTAMP()	129
MAKEDATE()	130
MAKETIME()	130
MICROSECOND()	130
MINUTE()	131
MONTH()	131
MONTHNAME()	131
NOW()	132
PERIOD_ADD()	132
PERIOD_DIFF()	133

QUARTER()	133
SEC_TO_TIME()	134
SECOND()	135
STR_TO_DATE()	135
SUBDATE()	135
SUBTIME()	136
SYSDATE()	136
TIME()	137
TIMEDIFF()	137
TIMESTAMP()	137
TIMESTAMPDIFF()	138
TIMESTAMPADD()	138
TIME_FORMAT()	138
TIME_TO_SEC()	139
TO_DAYS()	139
UNIX_TIMESTAMP()	139
UTC_DATE()	140
UTC_TIME()	140
UTC_TIMESTAMP()	140
WEEK()	141
WEEKDAY()	141
WEEKOFYEAR()	142
YEAR()	142
YEARWEEK()	142

7. Mathematical and Aggregate Functions 143

Functions in Alphabetical Order	143
ABS()	143
ACOS()	144
ASIN()	144
ATAN()	144
ATAN2()	144
AVG()	145
BIT_AND()	145
BIT_OR()	145
BIT_XOR()	145
CEIL()	145
CEILING()	146
COS()	146
COT()	146
COUNT()	146

CRC32()	147
DEGREES()	147
EXP()	147
FLOOR()	148
FORMAT()	148
GREATEST()	148
GROUP_CONCAT()	148
LEAST()	149
LN()	149
LOG()	150
LOG2()	150
LOG10()	150
MAX()	150
MIN()	150
MOD()	151
PI()	151
POW()	151
POWER()	152
RADIANS()	152
RAND()	152
ROUND()	152
SIGN()	153
SIN()	153
SQRT()	153
STD()	154
STDDEV()	154
SUM()	154
TAN()	154
TRUNCATE()	154
VARIANCE()	155
8. Flow Control Functions	156
Functions in Alphabetical Order	156
CASE()	156
IF()	157
IFNULL()	158
NULLIF()	159
9. Miscellaneous Functions	160
Functions in Alphabetical Order	160
ANALYSE()	160

BENCHMARK()	161
BIT_COUNT()	161
CAST()	162
CHARSET()	162
COALESCE()	162
COERCIBILITY()	162
COLLATION()	163
CONNECTION_ID()	163
CONVERT()	163
CURRENT_USER()	163
DATABASE()	164
FOUND_ROWS()	164
GET_LOCK()	164
INTERVAL()	165
IS_FREE_LOCK()	165
IS_USED_LOCK()	165
ISNULL()	166
LAST_INSERT_ID()	166
MASTER_POS_WAIT()	166
RELEASE_LOCK()	167
SESSION_USER()	167
SYSTEM_USER()	167
USER()	167
UUID ()	168
VERSION()	168
10. MySQL Server and Client	169
mysql	169
mysqld	171
mysqld_multi	177
mysqld_safe	178
11. Command-Line Utilities	180
comp_err	180
isamchk	181
make_binary_distribution	181
msql2mysql	181
my_print_defaults	181
myisamchk	182
myisamlog	185
myisampack	186

mysqlaccess	186
mysqladmin	189
mysqlbinlog	191
mysqlbug	192
mysqlcheck	193
mysqldump	194
mysqldumpslow	198
mysqlhotcopy	199
mysqlimport	200
mysqlshow	202
perror	203

12. Perl API 204

Using Perl DBI with MySQL	204
Perl DBI Method and Function Reference	208
available_drivers()	209
begin_work()	209
bind_col()	209
bind_columns()	209
bind_param()	210
bind_param_array()	210
bind_param_inout()	211
can()	211
clone()	211
column_info()	211
commit()	211
connect()	211
connect_cached()	212
data_sources()	212
disconnect()	212
do()	213
dump_results()	213
err()	214
errstr()	214
execute()	214
execute_array()	214
execute_for_fetch()	215
fetch()	215
fetchall_arrayref()	215
fetchall_hashref()	215
fetchrow_array()	216

fetchrow_arrayref()	216
fetchrow_hashref()	216
finish()	217
foreign_key_info()	217
func()	217
get_info()	217
installed_versions()	217
last_insert_id()	218
looks_like_number()	218
neat()	218
neat_list()	218
parse_dsn()	219
parse_trace_flag()	219
parse_trace_flags()	219
ping()	219
prepare()	219
prepare_cached()	220
primary_key()	220
primary_key_info()	220
quote()	220
quote_identifier()	221
rollback()	221
rows()	221
selectall_arrayref()	222
selectall_hashref()	222
selectcol_arrayref()	222
selectrow_array()	223
selectrow_arrayref()	223
selectrow_hashref()	224
set_err()	224
state()	224
table_info()	224
table_info_all()	225
tables()	225
trace()	225
trace_msg()	225
type_info()	225
type_info_all()	225
Attributes for Handles	226

13. PHP API	230
Using PHP with MySQL	230
PHP MySQL Functions in Alphabetical Order	232
mysql_affected_rows()	232
mysql_change_user()	233
mysql_client_encoding()	233
mysql_close()	233
mysql_connect()	234
mysql_create_db()	235
mysql_data_seek()	235
mysql_db_name()	236
mysql_db_query()	237
mysql_drop_db()	237
mysql_errno()	237
mysql_error()	238
mysql_escape_string()	238
mysql_fetch_array()	238
mysql_fetch_assoc()	239
mysql_fetch_field()	239
mysql_fetch_lengths()	240
mysql_fetch_object()	241
mysql_fetch_row()	241
mysql_field_flags()	242
mysql_field_len()	243
mysql_field_name()	243
mysql_field_seek()	244
mysql_field_table()	244
mysql_field_type()	245
mysql_free_result()	245
mysql_get_client_info()	245
mysql_get_host_info()	246
mysql_get_proto_info()	246
mysql_get_server_info()	246
mysql_info()	246
mysql_insert_id()	247
mysql_list_dbs()	247
mysql_list_fields()	247
mysql_list_processes()	248
mysql_list_tables()	248
mysql_num_fields()	249
mysql_num_rows()	249
mysql_pconnect()	249

mysql_ping()	250
mysql_query()	250
mysql_real_escape_string()	251
mysql_result()	251
mysql_select_db()	251
mysql_stat()	252
mysql_tablename()	252
mysql_thread_id()	252
mysql_unbuffered_query()	253
14. C API	254
Using C with MySQL	254
Functions in Alphabetical Order	257
mysql_affected_rows()	257
mysql_autocommit()	257
mysql_change_user()	258
mysql_character_set_name()	258
mysql_close()	258
mysql_commit()	259
mysql_connect()	259
mysql_create_db()	260
mysql_data_seek()	260
mysql_debug()	260
mysql_drop_db()	261
mysql_dump_debug_info()	261
mysql_eof()	261
mysql_errno()	261
mysql_error()	262
mysql_escape_string()	262
mysql_fetch_field()	263
mysql_fetch_field_direct()	263
mysql_fetch_fields()	264
mysql_fetch_lengths()	264
mysql_fetch_row()	265
mysql_field_count()	265
mysql_field_seek()	266
mysql_field_tell()	266
mysql_free_result()	267
mysql_get_client_info()	267
mysql_get_client_version()	267
mysql_get_host_info()	267
mysql_get_proto_info()	268

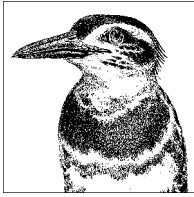
mysql_get_server_info()	268
mysql_get_server_version()	268
mysql_info()	269
mysql_init()	269
mysql_insert_id()	270
mysql_kill()	270
mysql_list_dbs()	270
mysql_list_fields()	271
mysql_list_processes()	271
mysql_list_tables()	272
mysql_more_results()	272
mysql_next_result()	272
mysql_num_fields()	273
mysql_num_rows()	273
mysql_options()	273
mysql_ping()	274
mysql_query()	275
mysql_real_connect()	275
mysql_real_escape_string()	277
mysql_real_query()	277
mysql_reload()	278
mysql_rollback()	278
mysql_row_seek()	278
mysql_row_tell()	279
mysql_select_db()	279
mysql_set_server_option()	280
mysql_shutdown()	280
mysql_sqlstate()	280
mysql_stat()	280
mysql_store_result()	281
mysql_thread_id()	281
mysql_thread_safe()	281
mysql_use_result()	282
mysql_warning_count()	282
C API Datatypes	283

A. Datatypes 285

B. Operators 290

C. Environment Variables 293

Index 295



Preface

MySQL is the most popular open source database system available. Although it's free, it's still very dependable and fast, and is being employed increasingly in areas that used to be the province of Oracle or MS SQL Server. Thanks to a variety of utilities packaged with MySQL, administration is fairly effortless. And with its several application programming interfaces (APIs), it's easy to develop your own software to interface with MySQL.

This book provides a quick reference to MySQL statements and functions, the administrative utilities, and the most popular APIs. The first few chapters are designed to help you get started with MySQL. Each chapter on an API also starts with a tutorial.

When this book was written, Version 4.1 of MySQL was released, and early releases of the development Version of 5.0 were available but not yet stable. As a result, you will find mostly features from Version 4.x, along with some from Version 5.x, in this book. Features that appear only in newer versions are noted as such.

The Purpose of This Book

The purpose of this book is to provide a quick reference to:

- MySQL statements and functions
- The most popular APIs used to access MySQL databases
- Command-line options and configuration information for the MySQL server and utilities

Several chapters start with tutorials, but the central purpose of the book is to fill in the gaps for people who are already comfortable with relational databases.

The format that I've followed for a description of each statement or function is to move from curt memory-joggers to more leisurely explanations. If you know the function that you're looking up, but can't quite remember the syntax, you'll find

that first. If you need a bit more information to jog your memory or to clarify the possibilities available with the function, you can find this in the first sentence or so of the explanation. If you require more clarification concerning a function's use, you can continue with the slower-paced material that follows. Examples of usage are provided for most functions.

In summary, the goal is to be brief but fairly complete, and to increase the level of detail as you read on.

How This Book Is Organized

This book is broken up into fourteen chapters and three appendixes, as follows:

Chapter 1, *Introduction to MySQL*, explains the major components of MySQL and useful guidelines for getting information and dealing with the MySQL community.

Chapter 2, *Installing MySQL*, describes how to get MySQL running on all the systems supported by MySQL AB. It is necessary to read this chapter only if your system does not already have MySQL installed.

Chapter 3, *MySQL Basics*, introduces SQL and use of the *mysql* command-line utility. It is no replacement for learning SQL and relational database design, but can be useful to orient you.

Chapter 4, *SQL Statements*, is the major chapter in this book, a comprehensive listing of all SQL statements supported by MySQL and their subclauses.

Chapter 5, *String Functions*, covers SQL functions for manipulating text data.

Chapter 6, *Date and Time Functions*, covers SQL functions that manipulate the various data and time formats supported by MySQL.

Chapter 7, *Mathematical and Aggregate Functions*, covers mathematical SQL functions and functions used for combining information in columns, such as `MAX()` and `COUNT()`.

Chapter 8, *Flow Control Functions*, covers SQL functions, such as `CASE` and `IF`.

Chapter 9, *Miscellaneous Functions*, covers SQL functions that can be used to control or get information about the database engine, and do other miscellaneous tasks.

Chapter 10, *MySQL Server and Client*, covers the options for the *mysqld* daemon, the *mysql* command-line client, and related commands.

Chapter 11, *Command-Line Utilities*, covers other commands for administering MySQL and its data.

Chapter 12, *Perl API* presents the Perl DBI module, used to access MySQL databases from Perl.

Chapter 13, *PHP API*, presents the PHP functions used to query and manipulate MySQL databases.

Chapter 14, *C API*, covers the data types and functions provided by MySQL's basic C library.

Appendix A, *Datatypes*, lists all the data types supported by MySQL.

Appendix B, *Operators*, lists all MySQL operators, such as arithmetic signs and the LIKE and IS NULL comparison operators.

Appendix C, *Environment Variables*, lists the operating system's environment variables consulted by the MySQL server, client, and other utilities.

Conventions Used in This Book

The following typographical conventions are used in this book:

Plain text

Indicates menu titles, menu options, menu buttons, and keyboard accelerators (such as Alt and Ctrl).

Italic

Indicates new terms, URLs, email addresses, usernames, hostnames, file-names, file extensions, pathnames, directories, and utilities.

Constant width

Indicates elements of code, configuration options, variables, functions, modules, the contents of files, or the output from commands.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*MySQL in a Nutshell* by Russell J.T. Dyer. Copyright 2005 O'Reilly Media, Inc., 0-596-00789-2.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Request for Comments

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

The examples in this book are professionally written and have been tested, but that does not mean that they are guaranteed to be bug-free or to work correctly with your version and your platform's implementation of MySQL 2.1. If you have problems, find bugs, or have suggestions for future editions, please email them to:

bookquestions@oreilly.com

There's a web page for this book that lists errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/mysqlian>

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

<http://www.oreilly.com>

Safari Enabled

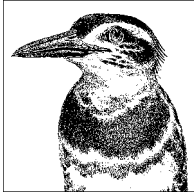


When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *<http://safari.oreilly.com>*.

Acknowledgments

Thanks to Andy Oram, my editor, for his guidance and editing, and for helping me to be the person fortunate enough to write this book. Thanks also to Kasia Trapszo, Rick Rezinaz, and others for reviewing the manuscript for technical accuracy. Their assistance is greatly appreciated. I also appreciate the chances that Rikki Endsley (editor of *Unix Review*) took with me at the start of my writing career, and the help that she, chromatic (editor of *ONLamp.com*), and several other magazines editors gave me in developing my skills. Special thanks to Kathryn Barrett (publicist for O'Reilly) for her moral support and advice over the past couple of years. Finally, thanks to my friends Rusty Osborne for listening to me through it all and Michael Zabalaoui for buying me lunch almost every weekday while I worked on this book in lieu of a regular job.



Introduction to MySQL

MySQL is an open source, multithreaded, relational database management system created by Michael “Monty” Widenius in 1995. In 2000, MySQL was released under a dual-license model that permitted the public to use it for free under the GNU Public License (GPL); this caused its popularity to soar. The company that owns and develops MySQL is MySQL AB (the AB stands for *aktiebolag*, or stock company). Currently, MySQL AB estimates that there are more than 4 million installations of MySQL worldwide, and reports an average of 35,000 downloads a day of MySQL installation software from its site and from mirror sites. The success of MySQL as a leading database is due not only to its price—after all, other cost-free and open source databases are available—but also its reliability, performance, and features.

The Value of MySQL

Many features contribute to MySQL’s standing as a superb database system. Its speed is one of its most prominent features. In a comparison by *eWEEK* of several databases—including MySQL, Oracle, MS SQL, IBM DB2, and Sybase ASE—MySQL and Oracle tied for best performance and for greatest scalability (see <http://www.mysql.com/it-resources/benchmarks> for more details). For a database long dismissed by many people, MySQL is remarkably scalable, and is able to handle tens of thousands of tables and billions of rows of data. Plus, it manages small amounts of data quickly and smoothly.

The storage engine, which manages queries and interfaces between a user’s SQL commands and the database’s backend storage, is the critical software in any database management system. MySQL offers several storage engines—previously called *table types*—with different advantages. Some are transaction-safe storage engines that allow for rollback of data. Additionally, MySQL has a tremendous number of built-in functions that are detailed in several chapters of this book.

MySQL is also very well-known for rapid and stable improvements. Whenever you visit MySQL AB's site to download MySQL, you will see a stable release that has been thoroughly tested. You will also see a distribution that has undergone testing, but contains components that have not been tested as thoroughly as the standard version. This version contains everything in the latest standard version plus new features that eventually will be rolled into the standard version. Each new release comes with speed and stability improvements, as well as new features.

The MySQL Package

The MySQL package comes with several programs. Foremost is the MySQL server, represented by the *mysqld* daemon. The daemon listens for requests on a particular port (3306 by default) by which clients submit queries. The standard MySQL client program is simply called *mysql*. With this text-based interface, a user can log in and execute SQL queries. This client can also accept queries from text files containing queries, and thereby execute them on behalf of the user or other software. However, most MySQL use is done by programs using a variety of languages. The interfaces for Perl, PHP, and C are discussed in this book.

A few wrapper scripts for *mysqld* come with MySQL. The *mysqld_safe* script is the most common way to start *mysqld*, because the script can restart the daemon if it crashes. This helps ensure minimal downtime for database services. The script *mysqld_multi* is used to start multiple sessions of *mysqld_safe*, and thereby multiple *mysqld* instances, for handling requests from different ports and different Unix socket files, and to make it easier to serve different sets of databases. For MS Windows NT and 2000 servers, there's *mysqld-nt*. It supports the named pipes that some Windows systems use instead of socket files.

MySQL also comes with a variety of utilities for managing a MySQL server. *mysqlaccess* is used for creating user accounts and setting their privileges. *mysqladmin* can be used to manage the MySQL server itself from the command line. This interaction includes checking a server's status and usage, and shutting down a server. *mysqlshow* may be used to examine a server's status, as well as information about databases and tables. Some of these utilities require Perl, or ActivePerl for Windows, to be installed on the server. See <http://www.perl.org> to download and install a copy of Perl on non-Windows systems, and <http://www.activestate.com/Products/ActivePerl> to download and install a copy of ActivePerl on Windows systems.

MySQL also comes with a few utilities for importing and exporting data from and to MySQL databases. *mysqldump* is the most popular for exporting data and table structures to a plain-text file known as a *dump* file. This can be used for backing up data or for manually moving it between servers. The *mysql* client can be used to import the data back to MySQL from a *dump* file.

mysqlhotcopy can also be used to back up a database or specific tables. It's more effective at data consistency between tables than *mysqldump*, because it locks the tables automatically. The resulting backup files are ready-to-use copies of the databases in the format MySQL uses. To restore them, you can just copy them to MySQL's data directory.

For importing data into MySQL from an external file that was exported in a common database format, MySQL provides *mysqlimport*.

Licensing

Although MySQL can be used for free and is open source, MySQL AB holds the copyrights to the source code. The company offers a dual-licensing program for its software: one allows cost-free use through the GPL under certain common circumstances, and the other is a commercial license bearing a fee. They're both the same software, but each has a different license and different privileges. See <http://www.fsf.org/licenses> for more details on the GPL.

MySQL AB allows you to use the software under the GPL if you use it without redistributing it, or if you redistribute it only with software licensed under the GPL. You can even use the GPL if you redistribute MySQL with software that you developed, as long as you distribute your software under the GPL as well.

However, if you have developed an application that requires MySQL for its functionality and you want to sell your software with MySQL under a nonfree license, you must purchase a commercial license from MySQL AB. There are other scenarios in which a commercial license may be required. For details on when you must purchase a license, see <http://www.mysql.com/company/legal/licensing>.

Besides holding the software copyrights, MySQL AB also holds the MySQL trademark. As a result, you cannot distribute software that includes MySQL in the name.

Mailing Lists

You can receive some assistance with problems that you may have with MySQL from the MySQL community at no charge through several listserv email systems hosted by MySQL AB. There is a main mailing list for MySQL (*mysql*) and several specialized mailing lists where anyone can post a message for help on a particular topic. One list covers questions about database performance (*benchmarks*). Another is for questions on the Windows versions of MySQL (*win32*). There are also lists for problems concerning the Java JDBC drivers (*java*) and for the Perl DBI module (*perl*).

For a complete listing or to subscribe to one or more of these mailing lists, go to <http://lists.mysql.com>. On this mailing list page, you will find links for subscribing to each list. When you click a subscription link, you will see a very simple form on which to enter your email address. Some subscribers, incidentally, like to use a special email address and name representing their online persona. It allows you anonymity and may make sorting emails easier. Others, however, prefer to use their real name and contact information. After you enter your email address, you will receive an automated message to confirm your address. That email will have a link to the MySQL site with some parameters identifying your address. Click the link, and it will open your web browser and confirm your subscription.

The page from which you can subscribe to a list also has links for unsubscribing from lists, as well as links to archives of previous listserv messages for each list. You can search these archives for messages from others who are describing the

same problem that you are trying to resolve. It's always a good idea to search archives before posting anything of your own, to find out whether your topic has been discussed before. If you can't find a solution in the documentation available to you or in the archives, you can post a message to a particular mailing list by sending an email to that list on *lists.mysql.com*. For example, if you have a problem with the Perl DBI module in relation to MySQL, you would send a message to *perl@lists.mysql.com*. Just be sure to send the message from the email account that is registered with the list to which you're submitting your question.

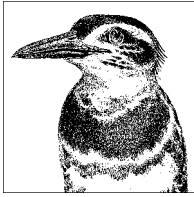
Books and Other Publications

Besides the mailing list archives mentioned in the last section, MySQL AB provides extensive online documentation of the MySQL server and all of the other software it distributes. You can find documentation at <http://dev.mysql.com/doc>. You can read the material online or download it in a couple of formats (e.g., HTML or PDF). It is also available in hardcopy format: *MySQL Language Reference* and *MySQL Administrator's Guide*, both from MySQL Press.

In addition to this book, O'Reilly Media publishes a few other books on MySQL worth buying and reading. O'Reilly's mainline MySQL book is *Managing & Using MySQL* (2002) by George Reese, Randy Jay Yarger, and Tim King (with Hugh E. Williams). George Reese has compiled a smaller version called *MySQL Pocket Reference* (2003). For common practical problem solving, there's *MySQL Cookbook* (2002) by Paul DuBois. For advice on optimizing MySQL and performing administrative tasks, such as backing up databases, O'Reilly has published *High Performance MySQL* (2004) by Jeremy D. Zawodny and Derek J. Balling.

O'Reilly also publishes several books with regard to the MySQL application programming interfaces (APIs). For PHP development with MySQL, there's *Web Database Applications with PHP and MySQL* (2004) by Hugh E. Williams and David Lane. For interfacing with Perl to MySQL and other database systems, there's *Programming the Perl DBI* (2000) by Alligator Descartes and Tim Bunce. To interface to MySQL with Java, you can use the JDBC and JConnector drivers and George Reese's book, *Database Programming with JDBC and Java* (2002).

In addition to published books on MySQL, a few web sites offer brief tutorials on using MySQL topics. The O'Reilly Network often publishes articles on MySQL and the APIs for Perl, PHP, and Python in its online publication ONLamp.com (<http://www.onlamp.com/onlamp/general/mysql.csp>). Incidentally, I've contributed a few articles to that publication on MySQL and related topics. I've also written many articles on MySQL for my column on Unix Review.com (<http://www.unixreview.com/mysql>). MySQL AB also provides some in-depth articles on MySQL. You can find them at <http://dev.mysql.com/tech-resources/articles>. Many of these articles deal with new products and features, making them ideal if you want to learn about using the latest releases available even while they're still in the testing stages. Developer Shed (<http://www.devarticles.com/cb/MySQL>) and Web Monkey (<http://search.hotwired.com/webmonkey/?query=mysql>) are additional educational resources. All of these online publications are subscription-free.



2

Installing MySQL

The MySQL database server and client software work on several different operating systems, notably Linux, FreeBSD, and a wide range of Unix systems: Sun Solaris, IBM AIX, HP-UX, and so on. MySQL AB has also developed a Mac OS X version, a Novell NetWare version, and several MS Windows versions. You can obtain a copy of MySQL from MySQL AB's site (<http://dev.mysql.com/downloads>) or from one of its mirror sites (<http://dev.mysql.com/downloads/mirrors.html>).

This chapter briefly explains the process of installing MySQL on Unix, Linux, Mac OS X, NetWare, and Windows operating systems. For some operating systems there are additional sections for different distribution formats. For any one platform, you can install MySQL by reading just three sections of this chapter: the next section, "Choosing a Distribution"; the section that applies to the distribution that you choose; and the "Postinstallation" section at the end of the chapter.

Choosing a Distribution

Before beginning to download an installation package, you must decide what version of MySQL to install. The best choice is usually the latest stable version recommended by MySQL AB on its site. It's not recommended that you install a newer version unless you need some new feature that is contained only in a newer version. It's also not recommended that you install an older version unless you have an existing database or an API application that won't function with the current version.

When installing MySQL, you also have the option of using either a source distribution or a binary distribution. It's easier, and recommended, for you to install a binary distribution. However, you may want to use a source distribution if special configuration must be performed during installation or at compile time. You may also have to use a source distribution if a binary distribution isn't available for your operating system.

For some distributions, you can download a Standard version, a Max version, or a Debug version. The Standard version is recommended for most users and developers, as it has been thoroughly tested.

Unix Source Distributions

The steps for installing MySQL on all Unix types of operating systems are basically the same. This includes Linux, Sun Solaris, FreeBSD, IBM AIX, HP-UX, etc. It's recommended that you install MySQL with a binary distribution, but as explained in the previous section, sometimes you may want to use a source distribution. To install a source distribution, you will need copies of GNU *gunzip*, GNU *tar*, GNU *gcc* (at least Version 2.95.2), and GNU *make*. These tools are usually included in all Linux systems and in most Unix systems. If your system doesn't have them, you can download them from the GNU Foundation's site (<http://www.gnu.org/>).

Once you've chosen and downloaded the source distribution files for MySQL, enter the following commands as *root* from the directory where you want the source files stored:

```
groupadd mysql
useradd -g mysql mysql
tar xvfz /tmp/mysql-version.tar.gz
cd mysql-version
```

The first command creates the user group *mysql*. The second creates the system user *mysql* and adds it to the group *mysql* at the same time. The next command uses the *tar* utility (along with *gunzip* via the *z* option) to unzip and unpack the source distribution file you downloaded. You should replace the word *version* with the version number—that is to say, you should use the actual path and filename of the installation file that you downloaded for the second argument of the *tar* command. The last command changes to the directory created by *tar* in the previous line. That directory contains the files needed to configure MySQL.

This brings you to the next step, which is to configure the source files to prepare them for building the binary programs. This is where you can add any special build requirements you may have. For instance, if you want to change the directory where MySQL is installed from the default, use the *--prefix* option with a value set to equal the desired directory. To set the Unix socket file's path, you can use *--with-unix-socket-path*. If you would like to use a different character set from the default of *latin1*, use *--with-charset*. Here is an example of how you might configure MySQL with these particular options before building the binary files:

```
./configure --prefix=/usr/local/mysql \  
            --with-unix-socket-path=/tmp \  
            --with-charset=latin2
```

You can also enter this command on one line without the backslashes.

Several other configuration options are available. To get a complete and current listing of options permitted, enter the following from the command line:

```
./configure --help
```

You may also want to look at the latest online documentation for compiling MySQL at http://dev.mysql.com/doc/mysql/en/Compilation_problems.html.

Once you've decided on any options that you want, run the configure script with these options. It will take quite a while to run, and it will display a great deal of information, which you can ignore usually if it ends successfully. After the configure script finishes, the binaries will need to be built and MySQL needs to be initialized. To do this, enter the following:

```
make
make install
cd /usr/local/mysql
./scripts/mysql_install_db
```

The first command builds the binary programs. If it's successful, you need to enter the second line to install the binary programs and related files in the appropriate directories. In the next line, you're changing to the directory where MySQL was installed. If you configured MySQL to be installed in a different directory, you'll have to use that one instead. The last command uses a script provided with the distribution to generate the initial privileges or grant tables.

All that remains now is to change the ownership of the MySQL programs and directories. You can do this by entering the following:

```
chown -R mysql /usr/local/mysql
chgrp -R mysql /usr/local/mysql
```

The first command changes ownership of the MySQL directories and programs to the *mysql* user. The second command changes the group owner of the same directory and files to *mysql*. These file paths may be different depending on the version of MySQL you installed and whether you configured MySQL for different paths.

With the programs installed and their file ownerships properly set, you can start MySQL. You can do this in several ways. To make sure that the daemon is restarted in the event that it crashes, enter the following from the command line:

```
/usr/local/mysql/bin/mysqld_safe &
```

This starts the *mysqld_safe* daemon, which will in turn start the MySQL server *mysqld*. If the *mysqld* daemon crashes, *mysqld_safe* will restart it. The ampersand at the end of the line instructs the shell to run the daemon in the background.

To have MySQL started at boot time, copy the *mysql.server* file, located in the *support-files* subdirectory of */usr/local/mysql*, to the */etc/init.d* directory. To do this, enter the following from the command line:

```
cp support-files/mysql.server /etc/init.d/mysql
chmod +x /etc/init.d/mysql
chkconfig --add mysql
```

The first line follows a convention of placing the startup file for the server in the server's initial daemons directory with the name *mysql*. The second command makes the file executable. The third sets the run level of the service for startup and shutdown.

Now that MySQL is installed and running, you need to make some postinstallation adjustments that are explained in the last section of this chapter.

Unix Binary Distributions

Installing MySQL with a binary distribution is easier than using a source distribution and is the recommended choice if a binary distribution is available for your platform. The files are packaged together into an archive file and then compressed before being placed on the Internet for downloading. Therefore, you will need a copy of GNU *tar* and GNU *gunzip* to be able to unpack the installation files. These tools are usually included on all Linux systems and most Unix systems. If your system doesn't have them, though, you can download them from the GNU Foundation's site (<http://www.gnu.org>).

Once you've chosen and downloaded the installation package, enter something like the following from the command line as *root* to begin the MySQL installation process:

```
groupadd mysql
useradd -g mysql mysql
cd /usr/local
tar xvfz /tmp/mysql-version.tar.gz
```

The first command creates the user group *mysql*. The second creates the user *mysql* and adds it to the group *mysql* at the same time. The next command changes to the directory where the MySQL files are about to be extracted. In the last command, you use the *tar* utility (along with *gunzip* via the *z* option) to unzip and unpack the source distribution file that you downloaded. The word “version” in the name of the installation file is replaced with the version number—that is to say, use the actual path and name of the installation file that you downloaded as the second argument of the *tar* command. For Sun Solaris systems, you should use *gtar* instead of *tar*.

After running the previous commands, you need to create a symbolic link to the directory created by *tar* in */usr/local*:

```
ln -s /usr/local/mysql-version /usr/local/mysql
```

This creates */usr/local/mysql* as a link to */usr/local/mysql-version*, where *mysql-version* is the actual name of the subdirectory that *tar* created in */usr/local*. The link is necessary, because MySQL is expecting the software to be located in */usr/local/mysql* and the data to be in */usr/local/mysql/data* by default.

At this point, MySQL is basically installed. Now you must generate the initial privileges or grant tables, and change the file ownership of the MySQL programs and datafiles. To do these tasks, enter the following from the command line:

```
cd /usr/local/mysql
./scripts/mysql_install_db
chown -R mysql /usr/local/mysql
chgrp -R mysql /usr/local/mysql
```

The first command changes to the directory containing MySQL's files. The second command uses a script provided with the distribution to generate the initial privileges or grant tables, which consist of the *mysql* database with MySQL's *root* user. The third command changes the ownership of the MySQL directories and programs to the *mysql* user. The last command changes the group owner of the same directory and files to *mysql*.

With the programs installed and their ownerships properly set, you can start MySQL. This can be done in several ways. To make sure that the daemon is restarted in the event that it crashes, enter the following from the command line:

```
/usr/local/mysql/bin/mysqld_safe &
```

The *mysqld_safe* daemon, started by this command, will in turn start the MySQL server *mysqld*. If the *mysqld* daemon crashes, *mysqld_safe* will restart it. The ampersand at the end of the line instructs the shell to run the command in the background.

To have MySQL started at boot time, copy the *mysql.server* file located in the *support-files* subdirectory of */usr/local/mysql*, to the */etc/init.d* directory. To do this, enter the following from the command line:

```
cp support-files/mysql.server /etc/init.d/mysql
chmod +x /etc/init.d/mysql
chkconfig --add mysql
```

The first line follows a convention of placing the startup file for the server in the server's initial daemons directory with the name *mysql*. The second command makes the file executable. The third sets the run level of the service for startup and shutdown.

Now that MySQL is installed and running, you need to make some postinstallation adjustments that are explained in the last section of this chapter.

Linux RPM Distributions

If your server is running on a version of Linux that installs software through the RPM package format (where RPM originally stood for RedHat Package Manager), it is recommended that you use a package instead of a source distribution. The differences between RPM versions are based not on the Linux distribution (e.g., SuSE or Mandrake), but on the Linux kernel or the type of libraries installed on the server. For each version of MySQL, there are a few RPM files that you can download. The primary two contain the server and client files. Their naming scheme is *MySQL-server-version.rpm* and *MySQL-client-version.rpm*, where *version* is the actual version number. In addition to these main packages, you may also want to install some of the other RPM files that are part of a distribution. There's an RPM for client-shared libraries (*MySQL-shared-version.rpm*), another for libraries and C API include files for certain clients (*MySQL-devel-version.rpm*), and another for benchmarking and other MySQL performance tests (*MySQL-bench-version.rpm*).

To install the RPM files after downloading them to your server, enter something like the following from the command line in the directory where they're located:

```
rpm -ivh MySQL-server-version.rpm \
MySQL-client-version.rpm
```

If an earlier version of MySQL is already installed on the server, you will receive an error message stating this problem, and the installation will be canceled. If you want to upgrade the existing installation, you can replace the *i* option in the example with an uppercase "U."