Updated for **MYSQL 8**

Ben Forta

MySQL®

CRASH COURSE

SECOND EDITION

GET UP AND RUNNING WITH MYSQL

MASTER MYSQL WORKBENCH

LEARN HOW TO RETRIEVE, SORT, AND FILTER DATA

TAKE ADVANTAGE OF REGULAR EXPRESSION FILTERING AND FULL TEXT SEARCHING

DISCOVER POWERFUL MYSQL FEATURES, INCLUDEING STORED PROCEDURES AND TRIGGERS

USE VIEWS AND CURSORS

MANAGE TRANSACTIONAL PROCESSING

CREATE USER ACCOUNTS AND MANAGE SECURITY VIA ACCESS CONTROL

FREE SAMPLE CHAPTER

MySQL Crash Course

This page intentionally left blank

MySQL Crash Course

Second Edition

Ben Forta

Addison-Wesley

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 the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2023943569

Copyright © 2024 Pearson Education, Inc.

Hoboken, New Jersey

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

ISBN-13: 978-0-13-822302-1
ISBN-10: 0-13-822302-5

$PrintCode

Vice President, IT Professional
Mark Taub

Acquisitions Editor
Kim Spenceley

Development Editor
Chris Zahn

Managing Editor
Sandra Schroeder

Senior Project Editor
Tonya Simpson

Copy Editor
Kitty Wilson

Indexer
Timothy Wright

Proofreader
Jennifer Hinchliffe

Editorial Assistant
Cindy Teeters

Cover Designer
Chuti Prasertsith

Compositor
codeMantra

Pearson's Commitment to Diversity, Equity, and Inclusion

Pearson is dedicated to creating bias-free content that reflects the diversity of all learners. We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, socioeconomic status, ability, age, sexual orientation, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where

- Everyone has an equitable and lifelong opportunity to succeed through learning
- Our educational products and services are inclusive and represent the rich diversity of learners
- Our educational content accurately reflects the histories and experiences of the learners we serve
- Our educational content prompts deeper discussions with learners and motivates them to expand their own learning (and worldview)

While we work hard to present unbiased content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.

Please contact us with concerns about any potential bias at
https://www.pearson.com/report-bias.html.

This page intentionally left blank

Contents

1 Understanding SQL 1

Database Basics 1

What Is a Database? 2

Tables 2

Columns and Datatypes 3

Rows 4

Primary Keys 4

What Is SQL? 6

Try It Yourself 6

Summary 7

2 Introducing MySQL 9

What Is MySQL? 9

Client/Server Software 9

MySQL Versions 10

MySQL Tools 11

`mysql` Command-Line Utility 11

MySQL Workbench 12

Other Tools 13

Summary 13

3 Working with MySQL 15

Using the Command-Line Tool 15

Selecting a Database 16

Learning About Databases and Tables 17

Using MySQL Workbench 20

Getting Started 20

Using MySQL Workbench 21

Selecting a Database 22

Learning About Databases and Tables 22

Executing SQL Statements 23

Next Steps 23

Summary 24

4 Retrieving Data 25

The SELECT Statement 25

Retrieving Individual Columns 25

Retrieving Multiple Columns 27
Retrieving All Columns 29
Retrieving Distinct Rows 29
Limiting Results 31
Using Fully Qualified Table Names 32
Using Comments 33
Summary 34
Challenges 34

5 Sorting Retrieved Data 35
Sorting Data 35
Sorting by Multiple Columns 37
Sorting by Column Position 38
Specifying Sort Direction 39
Summary 41
Challenges 42

6 Filtering Data 43
Using the `WHERE` Clause 43
`WHERE` Clause Operators 44
 Checking Against a Single Value 45
 Checking for Nonmatches 46
 Checking for a Range of Values 47
 Checking for No Value 48
Summary 49
Challenges 49

7 Advanced Data Filtering 51
Combining `WHERE` Clauses 51
 Using the `AND` Operator 51
 Using the `OR` Operator 52
 Understanding the Order of Evaluation 53
Using the `IN` Operator 54
Using the `NOT` Operator 56
Summary 58
Challenges 58

8 Using Wildcard Filtering 59

Using the LIKE Operator 59

The Percent Sign (%) Wildcard 60

The Underscore (_) Wildcard 61

Tips for Using Wildcards 63

Summary 63

Challenges 63

9 Searching Using Regular Expressions 65

Understanding Regular Expressions 65

Using MySQL Regular Expressions 66

Basic Character Matching 66

Performing OR Matches 68

Matching One of Several Characters 68

Matching Ranges 70

Matching Special Characters 70

Matching Character Classes 72

Matching Multiple Instances 72

Anchors 74

Summary 75

Challenges 76

10 Creating Calculated Fields 77

Understanding Calculated Fields 77

Concatenating Fields 78

Using Aliases 80

Performing Mathematical Calculations 81

Summary 83

Challenges 83

11 Using Data Manipulation Functions 85

Understanding Functions 85

Using Functions 86

Text Manipulation Functions 86

Date and Time Manipulation Functions 88

Numeric Manipulation Functions 91

Summary 92

Challenges 92

12 Summarizing Data 93
Using Aggregate Functions 93
The `Avg()` Function 94
The `Count()` Function 95
The `Max()` Function 96
The `Min()` Function 97
The `Sum()` Function 98
Aggregates on Distinct Values 99
Combining Aggregate Functions 100
Summary 101
Challenges 101

13 Grouping Data 103
Understanding Data Grouping 103
Creating Groups 104
Filtering Groups 105
Grouping and Sorting 107
Combining Grouping and Data Summarization 109
`SELECT` Clause Ordering 110
Summary 110
Challenges 110

14 Working with Subqueries 113
Understanding Subqueries 113
Filtering by Subquery 113
Using Subqueries As Calculated Fields 117
Summary 119
Challenges 119

15 Joining Tables 121
Understanding Joins 121
Understanding Relational Tables 121
Why Use Joins? 122
Creating a Join 123
The Importance of the `WHERE` Clause 124
Inner Joins 127
Joining Multiple Tables 128
Summary 130
Challenges 130

16 Creating Advanced Joins 133

Using Table Aliases 133

Using Different Join Types 134

Self-Joins 134

Natural Joins 136

Outer Joins 137

Using Joins with Aggregate Functions 138

Using Joins and Join Conditions 139

Summary 140

Challenges 140

17 Combining Queries 141

Understanding Combined Queries 141

Creating Combined Queries 141

Using `UNION` 141

`UNION` Rules 143

Including or Eliminating Duplicate Rows 144

Sorting Combined Query Results 145

Summary 146

Challenges 146

18 Full-Text Searching 147

Understanding Full-Text Searching 147

Using Full-Text Searching 148

Performing Full-Text Searches 148

Using Query Expansion 151

Boolean Text Searches 153

Full-Text Searching Notes 156

Summary 157

Challenges 157

19 Inserting Data 159

Understanding Data Insertion 159

Inserting Complete Rows 159

Inserting Multiple Rows 163

Inserting Retrieved Data 164

Summary 166

Challenges 166

20 Updating and Deleting Data 167
Updating Data 167
Deleting Data 169
Guidelines for Updating and Deleting Data 170
Summary 171
Challenges 171

21 Creating and Manipulating Tables 173
Creating Tables 173
Basic Table Creation 173
Working with `NULL` Values 175
Primary Keys Revisited 176
Using `AUTO_INCREMENT` 177
Specifying Default Values 178
Engine Types 179
Updating Tables 180
Deleting Tables 182
Renaming Tables 182
Summary 182
Challenges 182

22 Using Views 183
Understanding Views 183
Why Use Views 184
View Rules and Restrictions 185
Using Views 185
Using Views to Simplify Complex Joins 185
Using Views to Reformat Retrieved Data 186
Using Views to Filter Unwanted Data 188
Using Views with Calculated Fields 188
Updating Views 189
Summary 190
Challenges 190

23 Working with Stored Procedures 191
Understanding Stored Procedures 191
Why Use Stored Procedures 192
Using Stored Procedures 193

Executing Stored Procedures 193

Creating Stored Procedures 193

The DELIMITER Challenge 194

Dropping Stored Procedures 195

Working with Parameters 195

Building Intelligent Stored Procedures 199

Inspecting Stored Procedures 201

Summary 202

Challenges 202

24 Using Cursors 203

Understanding Cursors 203

Working with Cursors 204

Creating Cursors 204

Opening and Closing Cursors 205

Using Cursor Data 206

Summary 210

25 Using Triggers 211

Understanding Triggers 211

Creating Triggers 212

Dropping Triggers 213

Using Triggers 213

INSERT Triggers 213

DELETE Triggers 214

UPDATE Triggers 215

More on Triggers 216

Summary 216

26 Managing Transaction Processing 217

Understanding Transaction Processing 217

Controlling Transactions 219

Using ROLLBACK 219

Using COMMIT 220

Using Savepoints 220

Changing the Default Commit Behavior 221

Summary 222

27 Globalization and Localization 223

Understanding Character Sets and Collation Sequences 223

Working with Character Sets and Collation Sequences 224

Summary 226

28 Managing Security 227

Understanding Access Control 227

Managing Users 228

Creating User Accounts 229

Deleting User Accounts 230

Setting Access Rights 230

Changing Passwords 233

Summary 234

29 Database Maintenance 235

Backing Up Data 235

Performing Database Maintenance 235

Diagnosing Startup Problems 237

Reviewing Log Files 237

Summary 238

30 Improving Performance 239

Improving Performance 239

Summary 240

A Getting Started with MySQL 241

What You Need 241

Obtaining the Software 242

Installing the Software 242

Preparing to Read This Book 242

B The Example Tables 243

Understanding the Example Tables 243

Table Descriptions 244

The `vendors` Table 244

The `products` Table 244

The `customers` Table 245

The `orders` Table 245

The `orderitems` Table 246
The `productnotes` Table 246
Creating the Sample Tables 247
Using Data Import 247
Using SQL Scripts 248

C MySQL Statement Syntax 249
`ALTER TABLE` 249
`COMMIT` 249
`CREATE INDEX` 250
`CREATE PROCEDURE` 250
`CREATE TABLE` 250
`CREATE USER` 250
`CREATE VIEW` 251
`DELETE` 251
`DROP` 251
`INSERT` 251
`INSERT SELECT` 251
`ROLLBACK` 252
`SAVEPOINT` 252
`SELECT` 252
`START TRANSACTION` 252
`UPDATE` 252

D MySQL Datatypes 253
String Datatypes 253
Numeric Datatypes 255
Date and Time Datatypes 256
Binary Datatypes 256

E MySQL Reserved Words 257

Index 265

This page intentionally left blank

Acknowledgments

Thanks to the team at Pearson for all these years of support, dedication, and encouragement. Over the past two and a half decades, we've created 40+ books together, but our little *Sams Teach Yourself SQL in 10 Minutes* series remains my favorite by far. Thank you for trusting me with the creative freedom to evolve it as I have seen fit.

Speaking of *Sams Teach Yourself SQL in 10 Minutes*, that title covers MySQL (as it does all major DBMSs), but it cannot provide in-depth lessons on features that are truly unique to MySQL. This spinoff book was written in response to numerous requests from readers for greater MySQL-specific coverage. Thanks for the nudge. I hope this book lives up to your expectations.

Thanks to the many thousands of readers who provided feedback on prior editions of these books. Fortunately, most of it was positive; all of it was appreciated. The enhancements and changes in the latest editions are in direct response to your feedback, which I continue to welcome.

I write because I love to teach. While nothing compares to hands-on in-classroom instruction, turning those lessons into books that can be read far and wide has gifted me with expanding my teaching reach. It is thus a source of much gratification to see hundreds of colleges and universities use these SQL books as part of their IT and computer science curricula. Being included by professors and teachers in this way is both rewarding and humbling, and for that trust I am thankful.

And finally, thanks to the almost 1 million of you who bought the previous editions of these books (in over a dozen languages), making them not just my best-selling series but also the best-selling books on SQL. Your continued support is the highest compliment an author can ever be paid.

—Ben Forta

This page intentionally left blank

About the Author

Ben Forta is Adobe's Senior Director of Education Initiatives and has more than three decades of experience in the computer industry—in product development, support, training, and product marketing. He is the author of the best-selling *Sams Teach Yourself SQL in 10 Minutes* (as well as spinoff titles like this one and versions on SQL Server T-SQL, Oracle PL/SQL, and MariaDB), *Learning Regular Expressions*, and *Captain Code*, which teaches Python to younger coders (and those young at heart), Java, Windows, and more. He has extensive experience in database design and development, has implemented databases for several highly successful commercial software programs and websites, and is a frequent lecturer and columnist on application development and Internet technologies. Ben lives in Oak Park, Michigan, with his wife, Dr. Marcy Forta, and their children. He welcomes your email at ben@forta.com and invites you to visit his website at http://forta.com.

This page intentionally left blank

Introduction

MySQL is 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 for all sorts of data storage needs.

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 of the major database management systems (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, and transactions, in any real detail.

And that is where this book comes in. *MySQL Crash Course* builds on the proven tutorials and structure of *Sams Teach Yourself SQL in 10 Minutes* without getting bogged down with anything except MySQL. 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'll learn what you need to know methodically, systematically, and simply—in highly focused chapters designed to make you immediately and effortlessly productive.

When you turn to Chapter 1 and get to work, you'll be taking advantage of all MySQL has to offer in no time at all.

Who Is This Book For?

This book is for you if:

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

Companion Website

This book has a companion website online at http://forta.com/books/9780138223021/. At this website, you'll find:

- The files used to create the example tables used throughout this book
- Answers to the questions in the "Challenges" section at the end of each chapter
- Online errata

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 a placeholder with the specific value it represents.

Note

A Note presents an interesting piece 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

A New Term box provides a clear definition of a new essential term.

Figure Credits

Figures 3.1-3.5: Oracle Corporation

► 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 MySQL code. It usually appears after input and next to output.

► Analysis

The Analysis icon alerts you to the line-by-line analysis of input or output.

This page intentionally left blank

1

Understanding SQL

In this chapter, you'll learn about databases and SQL, which are prerequisites to learning MySQL.

Database Basics

The fact that you are reading this book indicates that you, somehow, need to interact with databases, and MySQL specifically. And so, before diving into MySQL and its implementation of the SQL language, it is important that you understand some basic concepts about databases and database technologies.

Whether you are aware of it or not, you use databases all the time. Each time you select a name from your email address book, you are using a database. When you browse contacts on your phone, you are using a database. If you conduct a search on an Internet search site, you are using a database. When you log in to your network at work, you are validating your name and password against a database. Even when you use your ATM card at a cash machine, you are using databases for PIN verification and balance checking.

But even though we all use databases all the time, there remains much confusion over what exactly a database is. This is especially true because different people use the same database terms to mean different things. Therefore, a good place to start our study is with a list and explanation of the most important database terms.

Tip

Reviewing Basic Concepts What follows is a very brief overview of some basic database concepts. It is intended to either jolt your memory if you already have some database experience or to provide you with the absolute basics if you are new to databases. Understanding databases is an important part of mastering MySQL, and you might want to find a good book on database fundamentals to brush up on the subject, if needed.

What Is a Database?

The term *database* is used in many different ways, but for our purposes in this book, a database is a collection of data stored in some organized fashion. The simplest way to think of it is to imagine a database as a filing cabinet. The filing cabinet is simply a physical location to store data, regardless of what that data is or how it is organized.

New Term

Database A container (usually a file or set of files) for storing organized data.

Caution

Misuse Causes Confusion People often use the term *database* to refer to the database software they are running. This is incorrect, and it is a source of much confusion. Database software is actually called a *database management system* (or *DBMS*). A database is a container created and manipulated via a DBMS. A database might or might not be a file stored on a hard drive. And for the most part, this is not even significant as you never access a database directly anyway; you always use the DBMS, and it accesses the database for you.

Tables

When you store information in a filing cabinet, you don't just toss it in a drawer. Rather, you create files within the filing cabinet, and then you store related data in specific files.

In the database world, a file is called a *table*. A table is a structured file that can store data of a specific type. A table might contain a list of customers, a product catalog, or any other list of information.

New Term

Table A structured list of data of a specific type.

The key here is that the data stored in the table is one type of data or one list. You would never store a list of customers and a list of orders in the same database table. Doing so would make subsequent retrieval and access difficult. Rather, you'd create two tables, one for each list.

Every table in a database has a name that identifies it. That name is always unique—meaning no other table in that database can have the same name.

Note

Table Names What makes a table name unique is actually a combination of several things, including the database name and table name. While you cannot use the same table name twice in the same database, you definitely can reuse table names in different databases.

Tables have characteristics and properties that define how data is stored in them. These include information about what data may be stored, how it is broken up, how individual pieces of information are named, and much more. The set of information that describes a table is known as a *schema*, and a schema can be used to describe specific tables within a database, as well as an entire database (and the relationship between tables in a database, if any).

New Term

Schema Information about database and table layout and properties.

Note

Schema or Database? Occasionally the term *schema* is used as a synonym for *database* (and *schemata* as a synonym for *databases*). While unfortunate and frequently confusing, it is usually clear from the context which meaning of schema is intended. In this book, *schema* is used as defined here.

Columns and Datatypes

Tables are made up of columns. A column contains a particular piece of information within a table.

New Term

Column A single field in a table. Every table is made up of one or more columns.

The best way to understand this is to envision database tables as grids, somewhat like spreadsheets. Each column in the grid contains a particular piece of information. In a customer table, for example, the customer number is stored in one column, the customer name is stored in another, and the address, city, state, and zip code are all stored in their own columns.

Tip

Breaking Up Data It is extremely important to break data into multiple columns correctly. For example, city, state, and zip code should always be stored in separate columns. By breaking these out, it becomes possible to sort or filter data by specific columns (for example, to find all customers in a particular state or in a particular city). If city and state are combined into one column, it would be extremely difficult to sort or filter by state.

Each column in a database has an associated datatype. A datatype defines what type of data the column can contain. For example, if a column is to contain a number (perhaps the number of items in an order), it would be associated with the numeric datatype.

Columns that contain dates, text, notes, currency amounts, and so on would use the appropriate datatypes.

New Term

Datatype A type of allowed data. Every table column has an associated datatype that restricts (or allows) specific data in that column.

Datatypes restrict the type of data that can be stored in a column (for example, preventing the entry of alphabetical characters into a numeric field). Datatypes also help sort data correctly and play an important role in optimizing disk usage. As such, special attention must be given to picking the right datatype when tables are created.

Rows

Data in a table is stored in rows; each record saved is stored in its own row. Again, if you envision a table as a spreadsheet-style grid, the vertical columns in the grid are the table columns, and the horizontal rows are the table rows.

For example, a customers table might store one customer per row. The number of rows in the table is the number of records in the table.

New Term

Row A record in a table.

Note

Records or Rows? You might hear users refer to database *records* when referring to *rows*. For the most part, the two terms are used interchangeably, but *row* is technically the correct term.

Primary Keys

Every row in a table should have some column (or set of columns) that uniquely identifies it. A table containing customers might use a customer number column for this purpose, whereas a table containing orders might use the order ID. Similarly, an employee list table might use an employee ID column.

New Term

Primary Key A column (or set of columns) whose values uniquely identify every row in a table.

The column (or set of columns) that uniquely identifies each row in a table is called a *primary key*. The primary key is used to refer to a specific row. Without a primary key, updating or deleting specific rows in a table is extremely difficult because there is no guaranteed safe way to refer to just the rows that are affected.

Tip

Always Define Primary Keys Although primary keys are not actually required, most database designers ensure that every table they create has a primary key so that future data manipulation is possible and manageable.

Any column in a table can be established as the primary key, as long as it meets the following conditions:

- No two rows can have the same primary key value.
- Every row must have a primary key value. (Primary key columns may not allow NULL values.)

Note

Primary Key Rules The rules listed here are enforced by MySQL itself.

A primary key is usually defined on a single column within a table. But this is not required, and multiple columns may be used together as a primary key. When multiple columns are used, the rules previously listed must apply to all columns that make up the primary key, and the values of all columns together must be unique. (Individual columns need not have unique values.)

Tip

Primary Key Best Practices In addition to following the rules that MySQL enforces, you should adhere to several universally accepted best practices:

- Don’t update values in primary key columns.
- Don’t reuse values in primary key columns.
- Don’t use values that might change in primary key columns. (For example, if you use a name as a primary key to identify a supplier and the supplier merges with another company and changes its name, you have to change the primary key.)

There is another very important type of key, called a *foreign key*, but I’ll get to that in Chapter 15, “Joining Tables.”

What Is SQL?

SQL (pronounced as the letters "S-Q-L" or as the word "sequel") is an abbreviation for Structured Query Language. SQL is a language designed specifically for communicating with databases.

Unlike other languages (spoken languages such as English or programming languages such as Python or Java), SQL is made up of very few words. This is deliberate. SQL is designed to do one thing and do it well: provide you with a simple and efficient way to read and write data from a database.

What are the advantages of SQL?

- SQL is not a proprietary language used by specific database vendors. Almost every major DBMS supports SQL, and learning this one language enables you to interact with just about every database you'll run into.
- SQL is easy to learn. The statements are all made up of descriptive English words, and there aren't that many of them.
- Despite its apparent simplicity, SQL is actually a very powerful language, and by cleverly using its language elements, you can perform very complex and sophisticated database operations.

Note

DBMS-Specific SQL Although SQL is not a proprietary language and there is a standards committee that tries to define SQL syntax that can be used by all DBMSs, the reality is that no two DBMSs implement SQL identically. The SQL taught in this book is specific to MySQL, and while much of the language taught will be usable with other DBMSs, you should not assume complete SQL syntax portability.

Try It Yourself

All of the chapters in this book use working examples, showing you the SQL syntax, what it does, and explaining why it does it. I strongly suggest that you try each and every example for yourself so that you learn MySQL firsthand.

In addition, starting in Chapter 4, "Retrieving Data," most chapters conclude with a "Challenges" section to help you review and gauge your MySQL proficiency. If you get stuck, you can go to the companion website to find the answers to the "Challenges" section questions.

Appendix B, "The Example Tables," describes the example tables used throughout this book and explains how to obtain and install them. If you have not done so yet, take a look at that appendix before proceeding.

Note

You Need MySQL Obviously, you'll need access to a copy of MySQL to follow along. Appendix A, "Getting Started with MySQL," explains where to get a copy of MySQL and provides some pointers for getting started. If you do not have access to a copy of MySQL, read that appendix before proceeding.

Summary

In this first chapter, you learned what SQL is and why it is useful. Because SQL is used to interact with databases, you also reviewed some basic database terminology.

This page intentionally left blank

Index

Symbols

! operator, 47
<> operator, 47

A

access control, 227–228, 230–233
administrative login, 15
Against() function, 148–151
aggregate functions, 93–94
  Avg(), 94–95
  combining, 100
  Count(), 95–96, 117
  DISTINCT argument. *See also* keywords and statements
  Max(), 96–97
  Min(), 97–98
  Sum(), 98–99
  using with joins, 138–139
algorithm, SOUNDEX, 87
aliases, 80–81, 100
ALTER TABLE statement, 180–181
ANALYZE TABLE statement, 236
anchors, 74–75
application filtering, 44
argument/s, 99–100. *See also* keywords and statements
AS keyword, 80–83, 133–134. *See also* joins and joining
ASC keyword, 40
AUTO_INCREMENT, 177–178
autocommit flag, 221–222
Avg() function, 94–95, 99–100

B

backing up data, 235
backslash (\ ), 71–72
best practices, primary key, 5
binary datatypes, 256
binary log, 237
Boolean text searches, 153–156

C

calculated fields, 77
  performing mathematical calculations, 81–82
  subqueries as, 117–119
  using with views, 188–189
calculations, testing, 82
CALL statement, 198
caret (^), 75
Cartesian product, 126
case
  converting, 86
  -sensitivity
    collocation, 223
    function, 80
    keyword, 27
    regular expression matching, 68
    search, 149
    SELECT statement, 226
    sort order and, 40
Cast() function, 226
changing user passwords, 233–234
character matching, 66–67

character classes, 72
escaping, 71
matching multiple instances, 72–74
matching one of several characters, 68–69
matching ranges, 70
OR operator, 68
special characters, 70–72
character sets, 223, 224–226
CHECK TABLE statement, 236
clause/s
FROM, 36
GROUP BY, 104–105
HAVING, 105–107
LIMIT, 41
ORDER BY, 36–37, 41
versus GROUP BY, 107–109
sorting by column position, 38
sorting by multiple columns, 37
sorting by nonselected columns, 37
specifying sort direction, 39–41
ordering, 110
VALUES, 160–161, 162
WHERE, 43–44
checking against a single value, 45–46
checking for nonmatches, 46–47
checking for NULL value, 48–49
filtering by date, 89–90
NOT operator, 56–57
AND operator, 51–52
OR operator, 52–53
IN operator, 54–56
operators, 44
using in joins, 124–127
client/server-based database, 9–10
closing
cursors, 205
implicit, 205
cloud-based DBMS, 10
code
looping, 208
portability, 85
collocation, 223, 224–226
specifying for a column, 225
table-wide, 224–225
column/s, 3–4
alias, 80–81
AUTO_INCREMENT, 177–178
collocation and character set, 225
datatype, 3–4
derived, 81
fully qualified names, 32–33, 124
individual, retrieving, 25–27
list, 161
listing, 17–19
nonselected, sorting by, 37, 38
omitting, 162
primary key, 4–5, 122, 176–177, 217
best practice, 5
rules, 5
retrieving all, 29
retrieving multiple, 27–28
setting the value to NULL, 169
sorting
by multiple, 37
by position, 38
unknown, retrieving, 29
updating, 168
combining, aggregate functions, 100
comma (,), 28
commands and command-line. *See also* keywords and statements
HELP SHOW, 20
keyword/s
ASC, 40
comments, 33–34
DESC, 40
presentation of data, 28
SELECT, 25
unsorted data, 35–36
wildcards, 29

mysql, 11, 16–17, 22
mysqld, 237
mysqldump, 235
mysqlhotcopy, 235
ROLLBACK, 219–220
selecting a database from, 16–17
semicolon (;), 26
SHOW, 17–20
white space, 27
COMMENT keyword, 200
comments, 33–34, 200
COMMIT statement, 220
compound queries, 141
creating using UNION keyword, 141–144
including or eliminating duplicate rows, 144–145
sorting query results, 145
Concat() function, 78–79
concatenating, fields, 78–80
conditions, join, 139–140
connecting to MySQL, selecting a database, 16–17
CONTINUE HANDLER, 207–208
Convert() function, 226
converting, case, 86
correlation subquery, 118
Count() function, 95–96, 117
CREATE PROCEDURE statement, 194
CREATE TABLE statement, 173–174
CREATE TRIGGER statement, 212
CREATE USER statement, 229
CREATE VIEW statement, 185
creating
compound queries, 141–144
cursors, 204
groups, 103–105
joins, 123–124
sample tables, 247–248
stored procedures, 193–194
tables, 173–174, 175
triggers, 212
user accounts, 229
cross join, 126
cursors, 203–204
closing, 205
creating, 204
declaring, 204
opening, 205
customers table, 245

D

data
backing up, 235
grouping, 103
importing from a table, 164–166
insertion. *See* INSERT statement
performing mathematical calculations, 81–82
reformatting, 186–187
removing, 169–170
sorting, 35–37
by column position, 38
direction, specifying, 39–41
by multiple columns, 37
by nonselected columns, 37
summarizing, 93–94, 109. *See also* aggregate functions; summarization
unwanted, filtering, 188
updating, 167–169, 180–181
database/s. *See also* table/s
client, 10
definition, 2
displaying details about, 22
listing, 17
maintenance, 235–236
ANALYZE TABLE statement, 236
CHECK TABLE statement, 236
mysql, user table, 228
reviewing log files, 237
scalability, 122
selecting

from command-line, 16–17
from MySQL Workbench, 22
server, 9–10
software, 2
table/s, 2–3
aliases, 133–134
columns, 3–4
primary key, 4–5
relational, 121–122
rows, 4
schema, 3
datatype/s, 3–4, 253
binary, 256
date and time, 256
numeric, 255–256
stored procedure parameter, 196
string, 253–254
date and time
datatypes, 256
manipulation functions, 88–90
Date(), 90–91
Month(), 91
Year(), 91
Date() function, 90–91
DBMS (database management system), 2, 9
client/server-based, 9–10
cloud-based, 10
shared file–based, 9
-specific SQL, 6
DECLARE statement, 204, 208
declaring, cursors, 203
default values, 178–179
DELETE statement, 169–170
DELETE trigger, 214–215
deleting
tables, 182
user account, 230
delimiter, 194–195
derived column, 81
DESC keyword, 40
DESCRIBE keyword, 19
diagnosing server startup problems, 237
dictionary sort order, 40
DISTINCT keyword, 29–30, 99–100
documentation, MySQL, 240
DROP PROCEDURE statement, 195
dropping
stored procedures, 195
triggers, 213

E

encoding, 223
engine types, 179–180
equijoin, 127
error log, 237
escaping, 71
example tables, 243–244
customers, 245
orderitems, 246
orders, 245–246
productnotes, 246–247
products, 244–245
vendors, 244
executing
SQL statements from MySQL Workbench, 23
stored procedures, 193

F

FETCH statement, 206–207
fields. *See also* columns
calculated, 77
performing mathematical calculations, 81–82
subqueries as, 117–119
using with views, 188–189
concatenating, 78–80
filter/ing. *See also* character matching; LIKE operator

application, 44
conditions, WHERE clause, 43–44
by date, 89–90
groups, 105–107
by more than one column, 51–52
regular expressions, 65–66
character matching, 66–67
escaping, 71
matching one of several characters, 68–69
matching ranges, 70
matching special characters, 70–72
pipe (|), 68
by subquery, 113–116
unwanted data, 188
FLUSH LOGS statement, 237
FLUSH TABLES statement, 235
foreign key, 122, 181
formatting
client versus server, 78
statements, 174
subqueries, 115
Forta, B., *Learning Regular Expressions*, 65
FROM clause, 36
full-text searches
Boolean mode, 153–156
case-sensitivity, 149
notes, 156–157
performing, 148–151
query expansion, 151–153
support, 147
fully qualified names, 32–33, 124
function/s, 85, 86
Against(), 148–151
aggregate, 93–94
Avg(), 94–95
combining, 100
Count(), 95–96, 117
DISTINCT argument, 99–100
Max(), 96–97
Min(), 97–98
Sum(), 98–99
using with joins, 138–139
case-sensitivity, 80
Cast(), 226
Concat(), 78–79
Convert(), 226
date and time manipulation, 88–90
Date(), 90–91
Month(), 91
Year(), 91
LTrim(), 80
Match(), 148–151
numeric manipulation, 91–92
portability, 85
RTrim(), 79
text manipulation, 87
Soundex(), 87–88
Upper(), 86

G-H

GRANT statement, 229, 231, 233
GROUP BY clause, 104–105, 107–109
groups and grouping, 1
combining with summarization, 109
creating, 103–105
filtering, 105–107

HAVING clause, 105–107
HELP SHOW command, 20

I

IF statement, 201
IGNORE keyword, 168–169
implicit closing, 205
importing, table data, 164–166
improving performance, 162, 239–240
inline comments, 33–34

inner join, 127
InnoDB, 179
INSERT SELECT statement, 164–166
INSERT statement, 159, 229
- inserting complete rows, 159–162
- inserting multiple rows, 163–164
- omitting columns, 162
- performance, 164
- syntax, 159–160

INSERT trigger, 213–214
inspecting stored procedures, 201
intelligent stored procedures, building, 199–201
IS NULL clause, 48–49

J

joins and joining, 121
- conditions, 139–140
- creating, 123–124
- cross, 126
- equi, 127
- importance of the WHERE clause, 124–127
- inner, 127
- multiple tables, 128–130
- natural, 136–137
- outer, 137–138
- reasons for using, 122–123
- self-, 134–136
- simplifying, 185–186
- table aliases, 133–134
- using with aggregate functions, 138–139

K

keywords and statements, see, 46–47. *See also* stored procedures
- ALTER TABLE, 180–181
- ANALYZE TABLE, 236
- AS, 80–83, 133–134
- ASC, 40
- AUTO_INCREMENT, 177–178
- CALL, 198
- case, 27
- CHECK TABLE, 236
- clauses
  - FROM, 36
  - LIMIT, 41
  - ORDER BY, 36–37
  - VALUES, 160–161, 162
- COMMENT, 200
- comments, 33–34
- COMMIT, 220
- CREATE PROCEDURE, 194
- CREATE TABLE, 173–174
- CREATE TRIGGER, 212
- CREATE USER, 229
- CREATE VIEW, 185
- DECLARE, 204, 208
- DELETE, 169–170
- DESC, 40
- DESCRIBE, 19
- DISTINCT, 29–30, 99–100
- DROP PROCEDURE statement, 195
- FETCH, 206–207
- FLUSH LOGS, 237
- FLUSH TABLES, 235
- formatting, 174
- GRANT, 229, 231, 233
- IF, 201
- IGNORE, 168–169
- INSERT, 159, 229
  - inserting complete rows, 159–162
  - inserting multiple rows, 163–164
  - omitting columns, 162
  - performance, 164
  - syntax, 159–160
- INSERT SELECT, 164–166
- LOOP, 208
- OPEN, 205
- OUT, 196

presentation of data, 28
REGEXP, 66–67
RENAME TABLE, 182
REPEAT, 206–209
reserved, 257–264
REVOKE, 231–232, 233
ROLLBACK, 219–220
ROLLUP, 105
SELECT, 25
  application filtering, 44
  combining aggregate functions, 100
  fully qualified names, 32–33
  LIMIT clause, 31–32
  LIMIT OFFSET clause, 32
  retrieving all columns, 29
  retrieving distinct rows, 29–30
  retrieving individual columns, 25–27
  retrieving multiple columns, 27–28
  unsorted data, 35–36
  WHERE clause, 43–44. *See also* WHERE clause
separating, 26
SET PASSWORD, 233–234
SHOW CHARACTER SET, 224
SHOW COLLOCATION, 224
SHOW GRANTS FOR, 230
SHOW PROCEDURE STATUS, 201
syntax. *See* syntax
triggers, 211, 216
  creating, 212–213
  DELETE, 214–215
  dropping, 213
  INSERT, 213–214
  multi-statement, 215
  UPDATE, 215–216
TRUNCATE TABLE, 170
UNION, 141–143
UNION ALL, 145
UPDATE, 167–169, 170
USE, 16–17, 22
white space, 27
wildcards, 29

L

language/s
  regular expression, 65
  SQL, 6
left outer join, 138
LIKE operator, 59
  percent sign (%) wildcard, 60–61
  versus REGEXP, 67
  underscore (_) wildcard, 61–62
LIMIT clause, 31–32, 41
LIMIT OFFSET clause, 32
log files, reviewing, 237
logging in, 15, 228
LOOP statement, 208
looping
  code, 208
  through cursor results, 206–207
LTrim() function, 80

M

maintenance. *See* database/s, maintenance
managing transactions, 219
  COMMIT statement, 220
  ROLLBACK statement, 219–220
  savepoints, 220–221
Match() function, 148–151
mathematical operators, 82
Max() function, 96–97
MEMORY engine, 179
metacharacters, 71
  anchor, 74–75
  repetition, 73
  white space, 71
Min() function, 97–98
Month() function, 91
multiline comments, 34
multi-statement triggers, 215
MyISAM, 179
MySQL, 9. *See also* commands and command-line

documentation, 240
engine
transaction support, 217
types, 179–180
logging in, 15
reserved words, 257–264
tools, 11, 13
mysql command-line utility, 11–12
MySQL Workbench, 12–13
versions, 10–11
mysql command-line utility, 11–12
specifying the user and password, 16
USE keyword, 16–17, 22
MySQL Workbench, 12–13
executing SQL statements, 23
MySQL Connections list, 20–21
Navigator, 22
New Query button, 23
password, 21
selecting a database, 22
user interface, 21–22
mysqld command, 237
mysqldump command, 235
mysqlhotcopy command, 235

N

name/s
alias, 80–81, 100
fully qualified, 32–33, 124
table, 2
variable, 197
natural joins, 136–137
Navigator, MySQL Workbench, 22
New Query button, 23
NOT operator, 56–57
NULL value, 175–176
checking for, 48–49
matching, 61
setting a column's value to, 169
numeric datatypes, 255–256
numeric manipulation functions, 91–92

O

OPEN statement, 205
BETWEEN operator, 47–48
AND operator, 51–52
OR operator, 52–53
IN operator, 54–56
OR operator, 68
BETWEEN operator, 91
operators
AND, 51–52
BETWEEN, 91
Boolean, 155
LIKE, 59
percent sign (%) wildcard, 60–61
versus REGEXP, 67
underscore (_) wildcard, 61–62
mathematical, 82
order of evaluation, 53–54
WHERE clause, 44
!, 47
<>, 47
BETWEEN, 47–48
IN, 54–56
NOT, 56–57
OR, 52–53, 68
Oracle, 11
ORDER BY clause, 36–37, 41
versus GROUP BY, 107–109
sorting by column position, 38
sorting by multiple columns, 37
sorting by nonselected columns, 37
specifying sort direction, 39–41
order of evaluation, 53–54
orderitems table, 246
orders table, 245–246

OUT keyword, 196
outer joins, 137–138

P

parameter/s
  datatypes, 196
  stored procedure, 195–199
parentheses, 54
password, 233
  changing, 233–234
  hashed, 229
  mysql command-line utility, 16
  MySQL Workbench, 21
percent sign (%) wildcard, 60–61
performance, 130
  full-text search, 147
  improving, 162, 239–240
  INSERT statement, 164
  joins, 129
  subqueries and, 116
  views, 184
period (.), in regular expressions, 67
pipe (|), 68
portability, function, 85
predicate, 60. *See also* operators
primary key, 4–5, 122, 176–177, 217
  best practice, 5
  rules, 5
productnotes table, 246–247
products table, 244–245

Q

query/ies. *See also* SELECT statement
  compound, 141
    creating using UNION keyword, 141–144
    including or eliminating duplicate rows, 144–145
    sorting the results, 145
  expansion, 151–153
  log, 237
question mark (?), 73
quotes, 46

R

ranges, matching, 70
records, 4
referential integrity, 123
reformatting retrieved data, 186–187
REGEXP keyword, 66–67
regular expressions, 65–66
  anchors, 74–75
  backslash (\ ), 71–72
  case-sensitivity, 68
  character classes, 72
  character matching, 66–67
  escaping, 71
  limitations, 147–148
  OR matches, 68
  matching multiple instances, 72–73
  matching one of several characters, 68–69
  matching ranges, 70
  matching special characters, 70–72
  period (.), 67
  testing, 75
relational tables, 121–122
removing, table data, 169–170
RENAME TABLE statement, 182
renaming
  tables, 182
  user account, 229–230
REPEAT statement, 206–209
repetition metacharacters, 73
reserved words, 257–264
reusable views, 186
REVOKE statement, 231–232, 233
right outer join, 138
rights and privileges
  user account, 232–233

ROLLBACK statement, 219–220
ROLLUP keyword, 105
root login, 228
rows, 4
  individual, retrieving, 29–30
  inserting, 159–162
  multiple, inserting, 163–164
RTrim() function, 79
rules
  primary key, 5
  UNION keyword, 143–144
  view, 185

S

sample tables, creating, 247–248
savepoints, 220–221
scalability, database, 122
schema, 3
scripts, 13
search pattern, 59
security, access control, 227–228
SELECT statement, 25. *See also* clause; keywords and statements; subquery/ies
  application filtering, 44
  calculated fields, 77
  case-sensitivity, 226
  clause ordering, 110
  combining aggregate functions, 100
  fully qualified names, 32–33
  GROUP BY clause, 104–105
  HAVING clause, 105–107
  LIMIT clause, 31–32
  retrieving all columns, 29
  retrieving distinct rows, 29–30
  retrieving individual columns, 25–27
  retrieving multiple columns, 27–28
  unsorted data, 35–36
  WHERE clause, 43–44
    checking against a single value, 45–46
    checking for nonmatches, 46–47
    operators, 44
    using in joins, 124–127
self-joins, 134–136
semicolon (;), 26, 27
separating multiple statements, 26
server
  database, 9–10
  diagnosing startup problems, 237
  formatting, 78
SET PASSWORD statement, 233–234
shared file–based DBMS, 9
SHOW CHARACTER SET statement, 224
SHOW COLLOCATION statement, 224
SHOW command, 17–20
SHOW GRANTS FOR, 230
SHOW PROCEDURE STATUS statement, 201
single quotes (‘ ’), 46
slow query log, 237
software, database, 2
sorting, 35–37
  case-sensitivity, 40
  by column position, 38
  compound query results, 145
  dictionary order, 40
  direction, specifying, 39–41
  by multiple columns, 37
  by nonselected columns, 37, 38
SOUNDEX, 87
Soundex() function, 87–88
special characters, matching, 70–72
SQL, 6
  DBMS-specific, 6
  executing statements, 23
square brackets ([])
  matching one of several characters, 68–69
  matching ranges, 70
statements. *See* keywords and statements
stopwords, 156
stored procedures, 191–192

calling another stored procedure, 209–210
comments, 200
creating, 193–194
with cursor, 206–208
delimiter, 194–195
dropping, 195
executing, 193
inspecting, 201
intelligent, 199–201
limiting status results, 201
looping, 206–207
opening and closing a cursor, 205–206
parameters, 195–199
reasons for using, 192–193
updating, 197
string datatypes, 253–254
subquery/ies, 113
as calculated fields, 117–119
correlation, 118
filtering by, 113–116
formatting, 115
order of processing, 115
performance and, 116
UPDATE statement, 169
Sum() function, 98–99
summarization, 93–94, 109
syntax
ALTER TABLE statement, 249
comment, 33–34
COMMIT statement, 249
CREATE INDEX statement, 250
CREATE PROCEDURE statement, 250
CREATE TABLE statement, 250
CREATE USER statement, 250
CREATE VIEW statement, 251
DELETE statement, 251
DROP statement, 251
inner join, 127
INSERT SELECT statement, 251
INSERT statement, 159–160, 251
IN operator, 55
ROLLBACK statement, 252
SAVEPOINT statement, 252
SELECT statement, 252
START TRANSACTION statement, 252
UPDATE statement, 168, 252

T

table/s, 2–3. *See also* views
aliases, 133–134
column/s, 3–4
alias, 80–81
AUTO_INCREMENT, 177–178
collocation, 225
datatype, 3–4
fully qualified names, 32–33
individual, retrieving, 25–27
listing, 17–19
omitting, 162
primary key, 4–5, 122, 176–177, 217
retrieving all, 29
retrieving multiple, 27–28
setting the value to NULL, 169
sorting by multiple, 37
sorting by position, 38
specifying a character set, 225
creating, 173–174, 175
deleting, 182
displaying details about, 22
example, 243–244
customers, 245
orderitems, 246
orders, 245–246
productnotes, 246–247
products, 244–245
vendors, 244
joins and joining, 122–123

creating, 123–124
cross, 126
importance of the WHERE clause, 124–127
inner, 127
multiple, 128–130
natural, 136–137
outer, 137–138
reasons for using, 122–123
self-, 134–136
listing, 17–19
names, 2
relational, 121–122
removing data from, 169–170
renaming, 182
rows, 4
individual, retrieving, 29–30
inserting, 159–162
multiple, inserting, 163–164
sample, creating, 247–248
schema, 3
specifying a character set and collocation, 224–225
updating data, 167–169, 180–181
user, 228
testing
calculations, 82
for equality, 44
queries, 119
regular expressions, 75
text manipulation functions, 87
Soundex(), 87–88
Upper(), 86
tools, 13. *See also* command/s
mysql command-line utility, 11–12
MySQL Workbench, 12–13
trailing spaces, wildcard, 61
transactions and transaction processing, 217–219
autocommit flag, 221–222
controlling, 219
COMMIT statement, 220
ROLLBACK statement, 219–220
savepoints, 220–221
triggers, 211, 216
creating, 212–213
DELETE, 214–215
dropping, 213
INSERT, 213–214
multi-statement, 215
UPDATE, 215–216
TRUNCATE TABLE statement, 170

U

underscore (_) wildcard, 61–62
UNION ALL keyword, 145
UNION keyword
compound queries, 141–143
including or eliminating duplicate rows, 144–145
rules, 143–144
unions. *See* compound queries
unknown columns, retrieving, 29
unsorted data, 35–36
unwanted data, filtering, 188
UPDATE statement, 167–169
guidelines, 170
IGNORE keyword, 168–169
subqueries, 168
views, 189–190
UPDATE trigger, 215–216
Upper() function, 86
USE keyword, 16–17, 22
user account
access control, 230–233
creating, 229
deleting, 230
password

changing, 233–234
hashed, 229
mysql command-line utility, 16
MySQL Workbench, 21
renaming, 229–230
rights and privileges, 232–233
specifying hashed password, 229
user interface, MySQL Workbench, 21–22
user table, 228

V

value/s
default, 178–179
deleting, 169
NULL, 175–176
checking for, 48–49
matching, 61
primary key, 4–5, 122, 176–177
best practice, 5
rules, 5
VALUES clause, 160–161, 162
variables, stored procedure, 197
vendors table, 244
versions, MySQL, 10–11
views, 183–184, 185
filtering unwanted data, 188
performance, 184
reasons for using, 184
reformatting retrieved data, 186–187
reusable, 186
rules and restrictions, 185
simplifying complex joins, 185–186
updating, 189–190
using with calculated fields, 188–189

W

WHERE clause, 43–44. *See also* filter/ing; regular expressions
checking against a single value, 45–46
checking for no value, 48–49
checking for nonmatches, 46–47
filtering by date, 89–90
NOT operator, 56–57
operators, 44
!, 47
<>, 47
AND, 51–52
BETWEEN, 47–48
IN, 54–56
OR, 52–53
order of evaluation, 53–54
parentheses, 54
subqueries, 113–116
using in joins, 124–127
white space
command-line, 27
metacharacters, 71
wildcard/s, 29, 59
limitations, 147–148
percent sign (%), 60–61
tips for using, 63
trailing spaces, 61
underscore (_), 61–62
using with DISTINCT argument, 100

X-Y-Z

Year() function, 91