Beginning MySQL®

Robert Sheldon and Geoff Moes

Wiley Publishing, Inc.
About the Authors

Robert Sheldon

Robert Sheldon’s MySQL programming is rooted in ten years of experience working with SQL, as it is implemented not only in a MySQL environment, but also within SQL Server, Microsoft Access, and Oracle environments. He has programmed with SQL directly through database interfaces and script files and indirectly through PHP, JSP, ASP, and ASP.NET applications that connected to various databases and issued SQL statements. Robert has also managed the maintenance and development of Web sites and online tools, which has included performing project analysis, developing functional specifications, and managing database and Web development. He has designed and implemented various Microsoft Access, SQL Server, and MySQL databases, as well as developed and implemented a variety of Web-based solutions. In all these roles, he has had to perform numerous types of ad hoc queries and modifications, build databases, create and modify database objects, create and review embedded statements, and troubleshoot system- and data-related problems.

In addition to having a technical and SQL background, Robert has written or co-written nine books on various network and server technologies, including two that have focused on SQL Server design and implementation, one on SQL programming (based on the SQL:1999 standard), and one on Microsoft Office Access 2003. The books that Robert has written contain training material that is designed to teach users specific skills and to test their knowledge of the material covered. Having contracted as the senior developmental editor for the Microsoft certification team, he brought to these books his experience developing exam items that helped to focus readers on the skills necessary to perform specific tasks. Robert has also written and edited a variety of other documentation related to SQL databases and other computer technologies. He works as an independent technical consultant and writer in the Seattle area.

Geoff Moes

Geoff Moes is a software architect and developer who has designed and implemented databases in MySQL as well as having designed and implemented software systems in PHP, Java/J2EE, and ASP.NET that have utilized MySQL databases through various database connectivity interfaces. Geoff received his bachelor’s degree in Computer Science from Virginia Tech and has worked in the software industry for 18 years. He specializes in software and database architecture and development as it relates to Web-based systems. He has worked with several database products in addition to MySQL, including SQL Server, DB2, and Oracle. He has also developed a variety of software applications that have connected to various databases using several different languages and platforms including Java J2EE/JDBC/EJB, C++/ODBC, and ASP.NET/ODBC/OLEDB.

Geoff’s publishing credits include “Passing Arrays Between Jscript and C++” (September 7, 2000, ASPToday.com, under WROX) and three articles published in Windows & .NET Magazine (online): “Common Internet Hacker Attacks” (December 1, 1998), “Remote Web Administration, Part 2” (November 1, 1998), and “Remote Web Administration, Part 1” (October 1, 1998). Geoff works as an independent software consultant in the Washington D.C. metro area. When he is not in front of the keyboard, he enjoys photography, mountain biking, hiking, and international travel.
Credits

Acquisitions Editor
Debra Williams Cauley

Development Editor
Brian Herrmann

Technical Editor
David Mercer

Copy Editor
Nancy Hannigan

Editorial Manager
Mary Beth Wakefield

Vice President and Publisher
Joseph B. Wikert

Project Coordinator
Erin Smith

Quality Control Technician
Brian H. Walls

Text Design and Composition
Wiley Composition Services

Proofreading and Indexing
TECHBOOKS Production Services

Vice President & Executive Group Publisher
Richard Swadley

Acknowledgments

As with any publication, too many people were involved in the development of Beginning MySQL to name them all, but we would like to acknowledge those who we worked with the closest in order to complete this project in a timely manner (and with our sanity still somewhat intact). Our special thanks goes to Debra Williams-Cauley, the acquisitions editor at John Wiley & Sons, Inc., who pulled this project together in such a professional and responsive manner. And we particularly want to thank Brian Herrmann, the development editor on this book who patiently and efficiently (and pleasantly, we might add) kept this project moving forward, while always paying attention to details and answering our never-ending stream of questions. We also want to acknowledge David Mercer, the technical editor, for his grasp of the subject matter and his invaluable input into the book. In addition, we want to acknowledge all the editors, proofreaders, indexers, designers, illustrators, and other participants whose efforts made this book possible. Finally, we want to thank our agent, Margot Maley Hutchison, at Waterside Productions, Inc., for her help in moving forward on this project and for tending to all the details.
# Contents

Acknowledgments v  
Introduction xxv  

## Chapter 1: Introducing the MySQL Relational Database Management System  

### Databases and Database Management Systems  
2  
What Is a Database?  
2  
The Hierarchical Model  
3  
The Network Model  
5  
The Relational Model  
6  
Database Management Systems  
7  
The MySQL RDBMS  
8  
The Open-Source Movement  
9  

### The SQL Framework  
9  
What is SQL?  
10  
A Brief History of SQL  
11  
The Nonprocedural Nature of SQL  
14  
SQL Statements  
14  
Types of Execution  
26  
Implementation-Specific SQL  
31  

### Data-Driven Applications  
32  
The Application Server  
32  
Connecting to a MySQL Database  
33  
Creating a Data-Driven Application  
35  

### Summary  
40  
Exercises  
42  

## Chapter 2: Installing MySQL  

### Getting Started  
43  
Making the Preliminary Decisions  
44  
Selecting a Version of MySQL  
44  
Selecting a Platform to Run MySQL  
44  
Selecting a MySQL Distribution Type  
45  
Selecting an Edition of MySQL  
46  
Downloading Distribution Files  
46
## Contents

### Installing MySQL
- Using RPM Files to Install MySQL on Linux  49
- Using a Tar File to Install MySQL on Linux  51
  - Creating the User and Group Accounts  51
  - Copying the Tar File  52
  - Unpacking the Distribution File  53
  - Initializing the Installation  54
  - Starting the MySQL Server  55
- Installing MySQL on Windows  57
  - Running the Installation Program  57
  - Configuring the MySQL Server  61
  - Editions of the MySQL Server  64

### Testing Your MySQL Installation
- Verifying Your Linux Installation  65
- Verifying Your Windows Installation  68

### Summary

### Exercises

---

### Chapter 3: Working with MySQL

#### Understanding the MySQL Directory Structure
- MySQL File Storage  74
  - File Storage for a Linux RPM Installation  74
  - File Storage for a Linux Tar Installation  75
  - File Storage for a Windows Installation  76
- The Data Directory  77
  - The mysql Database  78
  - The Grant Tables  79

#### Using the MySQL Programs
- Specifying Program Options  83
  - Specifying Options at a Command Prompt  83
  - Specifying Options in a Configuration File  85
- Server-Related Programs, Scripts, and Library Files  88
- Client Programs  89
  - The mysql Utility  90
    - Using mysql in Interactive Mode  90
    - Using mysql in Batch Mode  97

#### Assigning Account Passwords
- Setting Up a Configuration File  103

### Summary

### Exercises
Chapter 4: Designing a Relational Database 109

The Relational Model 109
Data Normalization 111
  First Normal Form 111
  Second Normal Form 113
  Third Normal Form 115
Relationships 116
  One-to-One Relationships 117
  One-to-Many Relationships 117
  Many-to-Many Relationships 119
Creating a Data Model 119
  Identifying Entities 122
  Normalizing Data 124
  Identifying Relationships 125
  Refining the Data Model 126
Designing the DVDRentals Database 127
Summary 136
Exercises 137

Chapter 5: Managing Databases, Tables, and Indexes 139

Managing Databases 139
  Creating Databases 140
  Modifying Databases 142
  Deleting Databases 142
Managing Tables 142
  Creating Tables 143
    Creating Column Definitions 144
    Defining a Column’s Nullability 152
    Defining Default Values 153
    Defining Primary Keys 154
    Defining Auto-Increment Columns 155
    Defining Foreign Keys 156
    Defining Table Types 158
  Creating Tables in the DVDRentals Database 160
  Modifying Tables 167
  Deleting Tables 169
Managing Indexes 171
  Index Types 171
  Creating Indexes 172
    Defining Indexes When Creating Tables 172
    Adding Indexes to Existing Tables 175
    Removing Indexes 179
## Contents

**Retrieving Information About Database Objects**  
Using SHOW Statements  
Using Database-Related SHOW Statements  
Using Table-Related SHOW Statements  
Using DESCRIBE Statements  
**Summary**  
**Exercises**

**Chapter 6: Manipulating Data in a MySQL Database**

**Inserting Data in a MySQL Database**  
Using an INSERT Statement to Add Data  
Using the `<values` option>` Alternative of the INSERT Statement  
Using the `<values` option>` Alternative to Insert Data in the DVDRentals Database  
Using the `<set` option>` Alternative of the INSERT Statement  
Using a REPLACE Statement to Add Data  
Using the `<values` option>` Alternative of the REPLACE Statement  
Using the `<set` option>` Alternative of the REPLACE Statement  
**Updating Data in a MySQL Database**  
Using an UPDATE Statement to Update a Single Table  
Using an UPDATE Statement to Update Joined Tables  
**Deleting Data from a MySQL Database**  
Using a DELETE Statement to Delete Data  
Deleting Data from a Single Table  
Deleting Data from Joined Tables  
Using a TRUNCATE Statement to Delete Data  
**Summary**  
**Exercises**

**Chapter 7: Retrieving Data from a MySQL Database**

**The SELECT Statement**  
Using Expressions in a SELECT Statement  
Using Variables in a SELECT Statement  
Using a SELECT Statement to Display Values  
**The SELECT Statement Options**  
**The Optional Clauses of a SELECT Statement**  
The WHERE Clause  
The GROUP BY Clause  
The HAVING Clause  
The ORDER BY Clause  
The LIMIT Clause  
**Summary**  
**Exercises**
## Contents

**Date/Time Functions** 339  
* ADDDATE(), DATE_ADD(), SUBDATE(), DATE_SUB(), and EXTRACT() Functions 339  
* CURDATE(), CURRENT_DATE(), CURTIME(), CURRENT_TIME(), CURRENT_TIMESTAMP(), and NOW() Functions 341  
* DATE(), MONTH(), MONTHNAME(), and YEAR() Functions 342  
* DATEDIFF() and TIMEDIFF() Functions 344  
* DAY(), DAYOFMONTH(), DAYNAME(), DAYOFWEEK(), and DAYOFYEAR() Functions 344  
* SECOND(), MINUTE(), HOUR(), and TIME() Functions 345  

### Summarizing Data 349  
**Summary Functions** 350  
* AVG() Function 350  
* SUM() Function 351  
* MIN() and MAX() Functions 352  
* COUNT() Function 353  

**Bit Functions** 354  

**Performing System Operations** 358  
**Encryption Functions** 358  
* ENCODE() and DECODE() Functions 358  
* PASSWORD(), MD5(), SHA(), and SHA1() Functions 359  

**System-Related Functions** 361  
* CURRENT_USER(), SESSION_USER(), SYSTEM_USER(), and USER() Functions 361  
* CONNECTION_ID(), DATABASE(), and VERSION() Functions 361  
* INET_ATON() and INET_NTOA() Functions 362  

**Query and Insert Functions** 362  
* FOUND_ROWS() Function 362  
* LAST_INSERT_ID() Function 363  

### Summary 366  

### Exercises 366  

**Chapter 10: Accessing Data in Multiple Tables** 369  

**Creating Joins in Your SQL Statements** 369  
* Joining Tables in a SELECT Statement 370  
* Creating Full Joins 373  
* Creating Outer Joins 385  
* Creating Natural Joins 391  
* Joining Tables in an UPDATE Statement 393  
* Joining Tables in a DELETE Statement 394  

**Creating Subqueries in Your SQL Statements** 396  
* Adding Subqueries to Your SELECT Statements 396  
* Working with Comparison Operators 396  
* Working with Subquery Operators 399  
* Working with Grouped Data 403
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Subqueries to Your UPDATE Statements</td>
<td>410</td>
</tr>
<tr>
<td>Adding Subqueries to Your DELETE Statements</td>
<td>411</td>
</tr>
<tr>
<td>Creating Unions That Join SELECT Statements</td>
<td>414</td>
</tr>
<tr>
<td>Summary</td>
<td>417</td>
</tr>
<tr>
<td>Exercises</td>
<td>418</td>
</tr>
<tr>
<td><strong>Chapter 11: Exporting, Copying, and Importing Data</strong></td>
<td>421</td>
</tr>
<tr>
<td>Exporting Data Out of a Table</td>
<td>421</td>
</tr>
<tr>
<td>Exporting Data to an Out File</td>
<td>423</td>
</tr>
<tr>
<td>Exporting Data to a Dump File</td>
<td>430</td>
</tr>
<tr>
<td>Copying Data into a Table</td>
<td>431</td>
</tr>
<tr>
<td>Copying Data into a New Table</td>
<td>432</td>
</tr>
<tr>
<td>Copying Data into an Existing Table</td>
<td>435</td>
</tr>
<tr>
<td>Using the INSERT Statement to Copy Data</td>
<td>435</td>
</tr>
<tr>
<td>Using the REPLACE Statement to Copy Data</td>
<td>436</td>
</tr>
<tr>
<td>Importing Data into a Table</td>
<td>438</td>
</tr>
<tr>
<td>Using the mysql Utility to Import Data</td>
<td>438</td>
</tr>
<tr>
<td>Using the LOAD DATA Statement to Import Data</td>
<td>438</td>
</tr>
<tr>
<td>Using the source Command to Import Data</td>
<td>444</td>
</tr>
<tr>
<td>Using the mysql Command to Import Data</td>
<td>446</td>
</tr>
<tr>
<td>Using the mysqlimport Utility to Import Data</td>
<td>449</td>
</tr>
<tr>
<td>Summary</td>
<td>452</td>
</tr>
<tr>
<td>Exercises</td>
<td>453</td>
</tr>
<tr>
<td><strong>Chapter 12: Managing Transactions</strong></td>
<td>455</td>
</tr>
<tr>
<td>Introducing Transactions</td>
<td>455</td>
</tr>
<tr>
<td>Performing a Transaction</td>
<td>457</td>
</tr>
<tr>
<td>Performing a Basic Transaction</td>
<td>457</td>
</tr>
<tr>
<td>The START TRANSACTION Statement</td>
<td>458</td>
</tr>
<tr>
<td>The COMMIT Statement</td>
<td>458</td>
</tr>
<tr>
<td>The ROLLBACK Statement</td>
<td>459</td>
</tr>
<tr>
<td>Statements That Automatically Commit Transactions</td>
<td>460</td>
</tr>
<tr>
<td>Adding Savepoints to Your Transaction</td>
<td>463</td>
</tr>
<tr>
<td>The SAVEPOINT Statement</td>
<td>464</td>
</tr>
<tr>
<td>The ROLLBACK TO SAVEPOINT Statement</td>
<td>465</td>
</tr>
<tr>
<td>Setting the Autocommit Mode and Transaction Isolation Level</td>
<td>468</td>
</tr>
<tr>
<td>Setting the Autocommit Mode</td>
<td>468</td>
</tr>
<tr>
<td>Setting the Transaction Isolation Level</td>
<td>472</td>
</tr>
<tr>
<td>Data Anomalies in a Transaction</td>
<td>472</td>
</tr>
<tr>
<td>Transaction Isolation Levels</td>
<td>475</td>
</tr>
<tr>
<td>Using the SET TRANSACTION Statement</td>
<td>476</td>
</tr>
</tbody>
</table>
## Contents

### Chapter 12: Extending MySQL

- Locking Nontransactional Tables 479
  - The LOCK TABLES Statement 480
  - The UNLOCK TABLES Statement 480
- Summary 483
- Exercises 483

### Chapter 13: Administering MySQL

- Performing Administrative Tasks 485
- Managing System Variables 494
  - Retrieving System Variable Settings 495
    - Using the SHOW VARIABLES Statement to Retrieve Server System Variable Settings 495
    - Using the SELECT Statement to Retrieve Dynamic System Variable Settings 497
    - Using the SHOW STATUS Statement to Retrieve Server Status Variable Settings 498
  - Modifying the Server Configuration 502
    - Specifying System Settings at the Command Line 502
    - Specifying System Settings in an Option File 503
    - Specifying System Settings at Runtime 503
- Managing Log Files 506
  - Working with Error Log Files 506
  - Enabling Query and Binary Logging 507
    - Setting Up Query Logging 507
    - Setting Up Binary Logging 507
- Summary 515
- Exercises 516

### Chapter 14: Managing MySQL Security

- The Access Privilege System 517
  - MySQL Grant Tables 518
    - The user Table 518
    - The db Table 520
    - The host Table 521
    - The tables_priv Table 522
    - The columns_priv Table 523
  - MySQL Privileges 524
- MySQL Access Control 527
  - Authenticating Connections 527
  - Verifying Privileges 529
- Managing MySQL User Accounts 533
  - Adding Users and Granting Privileges 533
    - Using the GRANT Statement 533
    - Using the SHOW GRANTS Statement 545
Contents

Managing Replication 610
  Managing the Master Server 611
  Managing the Slave Server 612
Summary 615
Exercises 616

Chapter 17: Connecting to MySQL from a PHP Application 617

Introduction to PHP 618
Building a Data-Driven PHP Application 619
  Connecting to a MySQL Database 619
  Retrieving Data from a MySQL Database 621
    Processing the Result Set 622
    Manipulating Data in PHP 623
    Converting Date Values 625
  Working with HTML Forms 626
  Redirecting Browsers 627
  Working with Include Files 627
  Creating a Basic PHP Application 628
  Inserting and Updating Data in a MySQL Database 639
  Adding Insert and Update Functionality to Your Application 640
  Deleting Data from a MySQL Database 660
Summary 664
Exercises 665

Chapter 18: Connecting to MySQL from a Java/J2EE Application 667

Introduction to Java/J2EE 667
Building a Java/J2EE Web Application 669
  Importing Java Classes 669
  Declaring and Initializing Variables 670
  Connecting to a MySQL Database 671
  Retrieving Data from a MySQL Database 672
    Processing the Result Set 673
    Manipulating String Data 674
  Converting Values 675
  Working with HTML Forms 677
  Redirecting Browsers 678
  Including JSP Files 678
  Managing Exceptions 679
  Creating a Basic Java/J2EE Web Application 679
  Inserting and Updating Data in a MySQL Database 692
  Adding Insert and Update Functionality to Your Application 693
  Deleting Data from a MySQL Database 716
## Contents

<table>
<thead>
<tr>
<th>Summary</th>
<th>721</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises</td>
<td>721</td>
</tr>
</tbody>
</table>

**Chapter 19: Connecting to MySQL from an ASP.NET/C# Application** 723

### Introduction to ASP.NET/C#

- Setting up the ASP.NET Web Page 725
- Declaring and Initializing Variables 725
- Connecting to a MySQL Database 727
- Retrieving Data from a MySQL Database 728
  - Processing the Result Set 729
  - Manipulating String Data 730
- Converting Values 731
- Working with HTML Forms 732
- Redirecting Browsers 733
- Including ASP.NET Files 733
- Managing Exceptions 734
- Creating a Basic ASP.NET/C# Application 735
- Inserting and Updating Data in a MySQL Database 748
- Adding Insert and Update Functionality to Your Application 750
- Deleting Data from a MySQL Database 773

### Summary 777

### Exercises 778

**Appendix A: Exercise Answers** 781

<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>781</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>781</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>782</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>782</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>782</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>782</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>783</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>783</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>784</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>784</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>785</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>785</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7</th>
<th>786</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>786</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8</th>
<th>787</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercises Solutions</td>
<td>787</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Exercises Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 9</td>
<td>788</td>
</tr>
<tr>
<td></td>
<td>788</td>
</tr>
<tr>
<td>Chapter 10</td>
<td>789</td>
</tr>
<tr>
<td></td>
<td>789</td>
</tr>
<tr>
<td>Chapter 11</td>
<td>790</td>
</tr>
<tr>
<td></td>
<td>790</td>
</tr>
<tr>
<td>Chapter 12</td>
<td>791</td>
</tr>
<tr>
<td></td>
<td>791</td>
</tr>
<tr>
<td>Chapter 13</td>
<td>792</td>
</tr>
<tr>
<td></td>
<td>792</td>
</tr>
<tr>
<td>Chapter 14</td>
<td>793</td>
</tr>
<tr>
<td></td>
<td>793</td>
</tr>
<tr>
<td>Chapter 15</td>
<td>794</td>
</tr>
<tr>
<td></td>
<td>794</td>
</tr>
<tr>
<td>Chapter 16</td>
<td>795</td>
</tr>
<tr>
<td></td>
<td>795</td>
</tr>
<tr>
<td>Chapter 17</td>
<td>797</td>
</tr>
<tr>
<td></td>
<td>797</td>
</tr>
<tr>
<td>Chapter 18</td>
<td>798</td>
</tr>
<tr>
<td></td>
<td>798</td>
</tr>
<tr>
<td>Chapter 19</td>
<td>799</td>
</tr>
<tr>
<td></td>
<td>799</td>
</tr>
</tbody>
</table>

Appendix B: MySQL APIs 801

Appendix C: MySQL 5 805

<table>
<thead>
<tr>
<th>Stored Procedures</th>
<th>Triggers</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>806</td>
<td>807</td>
<td>808</td>
</tr>
</tbody>
</table>

Index 811
Welcome to *Beginning MySQL*, the definitive resource for anyone new to the MySQL database management system. As the most popular open source database system in the world, MySQL has gained not only recognition among its peers but a place of prominence in the worldwide technical industry, ensuring an ever-growing need for information and training on how to implement a MySQL database and access and manage data in that database.

Because of its ease of implementation, low overhead, reliability, and lower Total Cost of Ownership (TCO), MySQL has made remarkable inroads in the database management system market. As a result, the number of programmers who must connect to a MySQL database and embed SQL statements in their applications is growing steadily. There are now over five million MySQL installations worldwide, and that number is increasing rapidly. In addition, MySQL supports connectivity to numerous application languages and environments, including C, C++, PHP, ASP, ASP.NET, Java, Perl, C#, and Python, and it can be implemented on a number of platforms, including Windows, Linux, Unix, Solaris, FreeBSD, Mac OS, and HP-UX.

Corporate implementations continue to grow and include such companies as Yahoo!, Cox Communications, Google, Cisco, Texas Instruments, UPS, Sabre Holdings, HP, and the Associated Press. Even NASA and the U.S. Census Bureau have implemented MySQL solutions. MySQL has been proven to work in large deployments, while reducing system downtimes and administrative overhead and lowering hardware expenditures and licensing costs.

As organizations continue to seek ways to cut their TCO, MySQL will continue to gain in popularity—and its user-base will continue to grow. As a result, MySQL will gain further ground in becoming a prominent force in the industry. To meet this demand, *Beginning MySQL* provides you with a valuable resource and step-by-step learning tool that supplies you with the background information, examples, and hands-on exercises that you need to implement MySQL and manage data in its databases. Concepts are introduced in a logical manner, with each chapter building on the previous chapters. By the end of this book, you’ll have a complete foundation in MySQL, its implementation, and the methods necessary to connect to databases and manipulate data.

**Who This Book Is For**

Before beginning any book that covers a computer technology such as MySQL, it’s always useful to know who the book is intended for, what assumptions are made about your level of knowledge, and what system setup—if any—is required to perform the exercises in the book. *Beginning MySQL* is no exception. So before you delve into the book too deeply, take a closer look at each of these issues.

Because MySQL is such a robust, flexible, and easy-to-implement application, a beginner’s book about the product will benefit a wide audience, both at home and at the office. The primary audience for *Beginning MySQL* can be any of the following readers:

- Experienced PHP, Java, or ASP.NET programmers who are developing applications that access backend databases and who are new to MySQL
Introduction

- Experienced application programmers in any language who are new to MySQL and who want to better understand how to implement a MySQL database and use SQL as it is implemented in MySQL
- Experienced SQL programmers new to MySQL
- Experienced database designers, administrators, or implementers who are migrating to MySQL
- First-time SQL programmers who have no database experience
- First-time database designers, administrators, or implementers new to MySQL
- Users new to application programming and databases

In addition to the primary audiences, *Beginning MySQL* can be useful to the following readers:

- The home user who wants to create simple databases for such information stores as address books, CD collections, or recipes
- The home business owner who wants to create database applications for such tasks as managing customers and contacts, tracking inventories, or recording orders
- Managers and owners of small businesses who need database solutions that are both easy and inexpensive to implement
- Group managers in larger companies who need database solutions that meet immediate needs in their groups
- Directors, staff, or volunteers at nonprofit organizations who require database solutions that are simple and inexpensive to implement
- Any other individual who wants to learn how to create and manage a MySQL database that can support various data-driven applications

Nearly anyone new to MySQL will be able to benefit from *Beginning MySQL*. In addition, users who have had experience with earlier versions of MySQL or with other database products will be able to use the book to refresh and expand their skills.

To benefit from *Beginning MySQL*, you do not have to have a strong background in databases or any other computer technology. You should, however, have at least a basic understanding of the following:

- You should know to negotiate your way around your operating system environment. The book focuses on implementing MySQL on Linux and Windows, so whichever one you choose, you should know how to use that system to copy and access files, add programs, change system settings, or whatever tasks are common to your particular environment. If you’re using a Unix-like system other than Linux, you should find that much of the Linux-related information will apply to your system.
- You will need to know how to use your Web browser to access the Internet and download files and view information.
- You should know how to use a text editor to create and edit text files.

These requirements are all you really need to use *Beginning MySQL* successfully and learn about how to implement MySQL databases and manage data in those databases. For Chapters 17, 18, or 19, you should have at least basic knowledge of Web development techniques. These three chapters focus on developing a Web application that accesses data in a MySQL database. Chapter 17 covers PHP, Chapter 18 covers...
JSP/Java, and Chapter 19 covers ASP.NET/C#. Each chapter assumes that you have a basic knowledge of developing an application in that language and of Web development in general. If you’re new to these technologies and you want to build an application in one of these languages, it’s recommended that you first review documentation specific to that language and to Web development in general.

What This Book Covers

The book uses a task-oriented structure that allows you to work through the steps necessary to install MySQL 4.1 on Linux and Windows platforms, create and manage MySQL databases, query and manipulate data stored in those databases, administer the MySQL database management system, and connect to MySQL databases from your PHP, JSP/Java, and ASP.NET/C# applications.

The next section, which describes the book’s structure, provides additional details about the specifics of what the book covers.

How This Book Is Structured

*Beginning MySQL* provides you with an instructional tool that gives you a complete look at MySQL, how it is implemented, and how it is accessed from various programming languages. The book takes a task-oriented, step-by-step approach to explain concepts and demonstrate how those concepts are used in real-world situations.

The structure of *Beginning MySQL* supports the complete beginner (those new to databases and SQL) as well as those who are experienced with programming and other database products, but new to MySQL. The book provides the conceptual and background information necessary for all readers to understand individual topics, but each chapter is modular to support those readers who simply dip into different parts of the book to use it as a reference. For example, someone completely new to databases might read the book from cover to cover, applying information learned in one chapter to the material in the next chapter. On the other hand, an experienced PHP programmer might want to reference only the chapters related to SQL statements and PHP connectivity, without having to review chapters on database design or administration.

*Beginning MySQL* describes and demonstrates each step necessary to create a MySQL database and access and manage data in that database. Each chapter correlates with one of the primary tasks necessary to implement MySQL and to access data, either directly or through an application programming language. The goal of the book is to provide you with a complete learning experience.

In Chapters 1 through 4, you are introduced to MySQL and relational databases. You are shown the steps necessary to install MySQL on Windows and Linux, set up the initial MySQL configuration, and access the MySQL server. You are also shown where to find MySQL components on your system and what tools are available to access and manipulate data. Finally, you learn how to design a database that conforms to the relational model. From this information, you will be ready to build a database in which you can store and manage data.

Chapters 5 through 12 build on the concepts introduced to you in the first four chapters. These chapters describe how to create databases that store data and help to enforce the integrity of that data. You then learn how to insert data into those databases, update that data, and then delete the data. You also learn a variety of methods to retrieve data from the database so that you can display exactly the data you need and perform operations on that data. You are also shown the steps necessary to copy, import, and export data.
Introduction

In Chapters 13 through 16, you learn how to perform tasks related to administering MySQL. The chapters include the steps necessary to perform such tasks as verify system settings and perform server-related operations, set up logging, manage security, optimize performance, back up and restore your system, and set up replication.

Chapters 17 through 19 are a little different from the other chapters. Each chapter describes how to access a MySQL database from a specific programming language, including PHP, JSP/Java, and ASP.NET/C#. You learn how to establish a connection to a database and issue SQL statements against the database. In each chapter, you will build a data-driven application that allows you to display data that you retrieve from a MySQL database.

In addition to the 19 chapters in this book, *Beginning MySQL* includes several appendices that provide additional information about MySQL and the book. Appendix A provides the answers to the exercises presented in each chapter. (The exercises are described in the text that follows.) Appendix B includes a brief description of each application programming interface (API) supported by MySQL, and Appendix C gives you an overview of features that you can expect to see in the next release of MySQL.

By the end of the book, you will have installed MySQL, configured it, created a database and its tables, added data to the database and manipulated that data, performed administrative tasks, and created applications that access the data in the database. To support this process, the chapters contain a number of elements, including examples that demonstrate how to perform various tasks, Try It Out sections that provide you with hands-on experience in using MySQL, and exercises that help you better understand the concepts explained in each chapter.

**Exercises and Examples**

To provide the various examples and exercises throughout the chapters, *Beginning MySQL* is based on MySQL 4.1. This means that the statement and command structures, various procedures, and expected results all conform to that version of the product. Specifically, most of what you find in the book applies to the MySQL 4.1.7 release or later. If you use another release of MySQL, however, you might find that some procedures and their results are different from what are described here.

As you work your way through the you’ll find that MySQL sometimes supports more than one way that an SQL statement can be written to achieve the same results. As MySQL has evolved, so too have the statements—to improve performance, simplify the statement, or conform to industry standards. The original version of the statements, however, continues to be maintained to support legacy MySQL systems or to provide portability from one database system to the next. In these situations, you should use whatever approach is best suited to your particular situation. This sometimes means trying different versions of a statement to determine which one performs the best. Often, you’ll find that using the simplest version not only is the easiest approach but will meet most of your needs.

Some of the code that you’ll run in the examples and Try It Out sections is available for download. Be sure to visit the Wrox Web site at [www.wrox.com](http://www.wrox.com) to determine whether code is available. Generally, anything that is more than a few lines (anything that you cannot easily type yourself) is available for download.

Each chapter in *Beginning MySQL* includes a number of elements that demonstrate how to perform specific tasks related to implementing MySQL databases and managing data in those databases. In addition to providing a thorough explanation of each concept, the chapters include examples, Try It Out sections, and exercises.
Examples

For each concept presented in the book, one or more examples are provided to demonstrate that concept. When appropriate, statement or command syntax precedes the examples. Syntax refers to the basic form that the statement or command should take. The syntax shows which elements must be included, which elements are optional, how parameters should be included, and in what order the elements should be placed. In other words, the syntax provides a blueprint for how a statement or command should be constructed. (Chapter 1 provides you with more information about syntax and supplies an example of how it is used.)

After the syntax has been provided, the chapter includes examples that demonstrate how real-life statements and commands are created. The examples are meant only as a way to demonstrate the actual code. You are not expected to try out the examples in an actual database environment. With the correct setup, the statements and commands do indeed work.

If you decide to try out an example, you can use the test database installed by default when you install MySQL, or you can create your own database for testing purposes. Keep in mind, however, that you have to pay particular attention to how the database is set up to try out the examples. You cannot assume that a table named Books used in the examples in one chapter is defined the same as a table named Books in another chapter. In addition, you cannot assume that, as examples progress through a chapter, they necessarily build on each other, although this is sometimes the case.

Whenever appropriate, a chapter will provide you with details about how the system is set up to demonstrate the examples. For instance, you’re often provided with the details about how a table is defined and what data it includes before the examples are provided. You can then use this table setup information to create your own table in order to try out an example. One example, however, can affect a table in such a way as to change its original structure or data, so you must be aware of this from one example to the next.

Again, the examples are meant only to demonstrate how a statement or command works. You’re not expected to try out each example, so if you do, pay close attention to your setup. The book also includes a number of Try It Out sections, which provide you with a more controlled environment to try out statements and commands.

Try It Out Sections

Most chapters include one or more Try It Out sections that provide the details necessary for you to try out the concepts explained in the chapter. Each Try It Out section contains steps that you should follow to perform specific tasks correctly. Each step explains what action you should take and, when applicable, provides the code that you should execute. At the end of each Try It Out section, you will find a How It Works section that explains in detail the steps that you took in the Try It Out section.

Many of the Try It Out sections build on each other as you progress through the book. For example, in Chapter 4 you design a database, in Chapter 5 you create a database based on that design, and in Chapter 6 you add data to the database. The same database is then used in most of the Try It Out sections that follow Chapter 6. In fact, you will use the same database to support the data-driven application that you create in Chapter 17, 18, or 19.

As you work your way through the book, you’ll also find that concepts introduced in earlier chapters and demonstrated in the related Try It Out sections are not explained in detail in later Try It Out sections. The assumption in the later Try It Out sections is that you performed the earlier exercises and now know the material.
Introduction

In general, you’ll find that the most effective way to use the Try It Out sections is to perform them in sequential order and make certain that you thoroughly understand them before moving on to the next set of Try It Out sections. By performing the exercises sequentially, you will have, by the end of the chapter, designed and created a database, added data to and manipulated data in that database, administered the MySQL server and the database, and built a PHP, JSP/Java, or ASP.NET/C# application that accesses data in the database.

Exercises

In addition to the examples and the Try It Out sections, each chapter ends with a set of exercises that allow you to further build on and test the knowledge that you acquired in that chapter. The answers to these exercises can be found in Appendix A. Keep in mind, however, that the answers provided for the exercises sometimes represent only one possible solution. As a result, you might come up with an answer that is also correct but different from what is shown in Appendix A. If this is the case, you can refer to the actual chapter content to confirm your answer. The answers shown in the appendix normally represent the most straightforward solution to the exercise, based on the information in the chapter.

Overall, you’ll find the exercises to be a useful tool to help better comprehend the concepts presented in the chapter. The exercises, along with the examples and Try it Out sections, provide you with a cohesive presentation of the material so that you can understand each concept completely. By taking advantage of each of these elements, you will have a thorough foundation of MySQL and will understand the steps necessary to install and implement MySQL and manipulate data in a MySQL database.

What You Need to Use This Book

Beginning MySQL contains numerous examples and exercises. If you plan to try out these exercises, you need a system on which to implement MySQL. Specifically, your system should meet the following requirements:

- You should be working on a computer that has a Windows or Linux operating system installed. You can usually substitute another Unix-like system for Linux, although some of the exercises might work a little differently from those that focus on Linux.
- Eventually, you will need to install the MySQL database management system on your computer. Chapter 2 explains how to install MySQL.
- If you plan to download MySQL or any other files from the Web, you will need high-speed Internet access.
- You will need a text editor such as Vim (for Linux) and Notepad (for Windows).
- For Chapters 17, 18, and 19, you will need the appropriate environment in which to implement your application. For example, PHP requires a Web server such as Apache. JSP/Java requires a Web server or application server such as JBoss. Depending on your JSP/Java Web server or application server, you might also need a special compiler. ASP.NET/C# requires a Web server such as Internet Information Services. In addition, regardless of the type of application you create, your system must be set up with the MySQL driver necessary to allow your application to connect to the MySQL server.

Once you have your system set up the way you need it, you’re ready to begin working your way through Beginning MySQL.
Conventions

To help you get the most from the text and keep track of what’s happening, a number of conventions are used throughout the book.

**Try It Out**

The *Try It Out* is an exercise you should work through, following the text in the book.

1. They usually consist of a set of steps.
2. Each step has a number.
3. Follow through the steps with your copy of the database.

**How It Works**

After each *Try It Out*, the code you’ve typed will be explained in detail.

*Tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.*

As for styles in the text:

- New terms and important words are *italicized* when they are introduced.
- Keyboard strokes are shown like this: Ctrl+A.
- File names, URLs, and code in the text are shown like so: *persistence.properties*.
- Code is presented in two different ways:

  In code examples we highlight new and important code with a gray background.

  The gray highlighting is not used for code that is less important in the present context or has been shown before.

**Source Code**

As you work through the examples in this book, you may choose either to type all the code manually or to use the source code files that accompany the book. Much of the source code used in this book is available for download at [http://www.wrox.com](http://www.wrox.com). (Generally, if the code in an example or in a step in a *Try It Out* section is only a few lines, that code is not included.) Once at the site, simply locate the book’s title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book’s detail page to obtain all the source code for the book.

*Because many books have similar titles, you may find it easiest to search by ISBN; for this book the ISBN is 0-764-57950-9.*

Once you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at [http://www.wrox.com/dynamic/books/download.aspx](http://www.wrox.com/dynamic/books/download.aspx) to see the code available for this book and all other Wrox books.
Introduction

Errata

We make every effort to ensure that there are no errors in the text or in the code. No one is perfect, though, and mistakes do occur. If you find an error in one of our books, such as a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and at the same time you will be helping us provide even higher-quality information.

To find the errata page for this book, go to http://www.wrox.com and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata submitted for this book and posted by Wrox editors. A complete book list including links to each's book's errata is also available at www.wrox.com/misc-pages/booklist.shtml.

If you don't spot "your" error on the Book Errata page, go to www.wrox.com/contact/techsupport.shtml and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at p2p.wrox.com. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At http://p2p.wrox.com you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to p2p.wrox.com and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P, but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.
Introducing the MySQL Relational Database Management System

In the world of online auction houses, instant mortgages, worldwide reservations, global communication, and overnight deliveries, it’s not surprising that even the least technically savvy individuals in our culture are, to some degree, familiar with the concept of a database. As anyone who works with data knows, databases form the backbone for this age of information, and access to those databases can determine one’s ability to perform critical tasks effectively and efficiently. To meet the ever-increasing demands for information, programmers are continuously building bigger and better applications that can access and modify data stored in various database systems. Yet in order to create these applications, programmers must have some knowledge of the systems that contain the needed data.

Over the years, as the demands for information have grown, so too have the database systems that have attempted to meet these demands. However, along with this evolution, we have seen an increase in the costs associated with storing data as well as an increase in the demand for products that can run on multiple platforms and can be optimized based on the needs of specific types of organizations. In response to this changing climate, MySQL has emerged as the most popular open-source database management system (DBMS) in the world. Consequently, organizations everywhere are jumping on the MySQL bandwagon, increasing the demand for those who know how to use MySQL to manage data and those who know how to create applications that can access data in MySQL databases.

In learning to use MySQL, whether to work directly in the MySQL environment or create data-driven applications, an individual should have a thorough understanding of how MySQL, as a relational database management system (RDBMS), allows you to manage data and support applications that rely on access to MySQL data. To this end, this chapter introduces you to MySQL and provides you with an overview of databases, RDBMSs, SQL, and data-driven applications. By the end of the chapter, you will understand the following concepts:
Databases and Database Management Systems

Databases and database management systems have become the backbone of most Web-related applications as well as an assortment of other types of applications and systems that rely on data stores to support dynamic information needs. Without the availability of flexible, scalable data sources, many organizations would come to a standstill, their ability to provide services, sell goods, rent movies, process orders, issue forms, lend books, plan events, admit patients, and book reservations undermined by the inability to access the data essential to conducting business. As a result, few lives are unaffected by the use of databases in one form or another, and their ubiquitous application in your everyday existence can only be expected to grow.

What Is a Database?

Over the years, the term database has been used to describe an assortment of products and systems that have included anything from a collection of files to a complex structure made up of user interfaces, data storage and access mechanisms, and client/server technologies. For example, a small company might store payroll records in individual files, while a regional electric company uses an integrated system to maintain records on all its customers; generate electric bills to those customers; and create reports that define power usage patterns, profit and loss statements, or changes in customer demographics. In both cases, the organizations might refer to each of their systems as databases.

Despite how a database is used, the amount of data that it stores, or the complexity of the data, a number of common elements define what a database is. At its simplest, a database is a collection of data that is usually related in some fashion. For instance, a database that a bookstore uses might contain information about authors, book titles, and publishers. Yet a database is more than simply a collection of related data. The data must be organized and classified in a structured format that is described by metadata, which is data that describes the data being stored. In other words, the metadata defines how the data is stored within the database. Together, the data and the metadata provide an environment that logically organizes the data in a way that can be efficiently maintained and accessed.

One way to better understand what constitutes a database is to use the analogy of a telephone book. A phone book contains the names, addresses, and phone numbers of most of the telephone customers in a particular town or region. If you think of that phone book as a database, you find a set of related data (the names, addresses, and phone numbers of the telephone customers) and you find a structured format (the metadata) that is defined by the way that the pages are bound and by how the information is organized. The phone book provides a system that allows you easy and efficient access to the data contained in its pages. Without the structure of the phone book, it would be next to impossible to locate specific customer data.
In the same way that a phone book provides structure to the customer information, the metadata of a database defines a structure that organizes data logically within that structure. However, not all database structures are the same, and through the years, a number of different data models have emerged. Of these various models, the three most commonly implemented are the hierarchical, network, and relational models.

The Hierarchical Model

In the early days of database design, one of the first data models to emerge was the hierarchical model. This model provides a simple structure in which individual records are organized in a parent-child relationship to form an inverted tree. The tree creates a hierarchical structure in which data is decomposed into logical categories and subcategories that use records to represent the logical units of data.

Take a look at an example to help illustrate how to structure a hierarchical database. Suppose you’re working with a database that stores parts information for a company that manufactures wind generators. Each model of wind generator is associated with a parent record. The parts that make up that model are then divided into categories that become child records of the model’s parent record, as shown in Figure 1-1. In this case, the parent record — Wind Generator Number 101 — is linked to three child records: Tower assemblies, Power assemblies, and Rotor assemblies. The child records are then divided into subcategories that are assigned their own child records. As a result, the original child records now act as parent records as well. For example, the Tower assemblies record is a parent of the Towers record but a child of the Wind Generator Number 101 record.

As you can see in the figure, a parent record can be associated with multiple child records, but a child record can be associated with only one parent record. This structure is similar to what you might see in a directory structure viewed through a graphical user interface (GUI) file management application, such as Windows Explorer. At the top of the directory structure would be Wind Generator Number 101. Beneath this, would be Tower assemblies, Power assemblies, and Rotor assemblies, each with their own set of subdirectories.

After its introduction, the hierarchical data model achieved a great deal of success. One of the most popular implementations of this model is found in IBM’s Information Management System (IMS), which was introduced in the 1960s and is still widely used on IBM mainframe computers.

However, despite the popularity of the hierarchical model, it is unsuitable for many of today’s applications. Inherent in the simplicity of the parent-child organization is a rigid structure that results in a cumbersome navigation process that requires application developers to programmatically navigate through the connected records to find the necessary information. Records must be accessed one at a time by moving up and down through the hierarchical levels, which often made modifications to the database and application a complicated, time-consuming process. In addition, the hierarchical structure cannot support complex relationships between records. For example, if you return to the wind generator database example, you discover that Figure 1-1 doesn’t show a record for the belts used to connect the generators to the shafts. If you were to create a child record for belts, should it be added under the Generators record or the Shaft assemblies record? The hierarchical design makes it difficult to fully represent the relationship that exists between the belts and the generators and shafts. Indeed, a child record can be associated with only one parent record.
Even with the limitations of the hierarchical model, a large amount of data is still being stored in hierarchical databases, and many management systems have found ways to work around some of these limitations. In addition, this is the type of system used primarily for file management systems associated with operating systems because it allows users to go directly where they need to go to find a file, rather than having to iterate through a lot of nodes. As a result, the hierarchical database probably isn’t going anywhere anytime soon.