

---

# Database Concepts

7th Edition

---

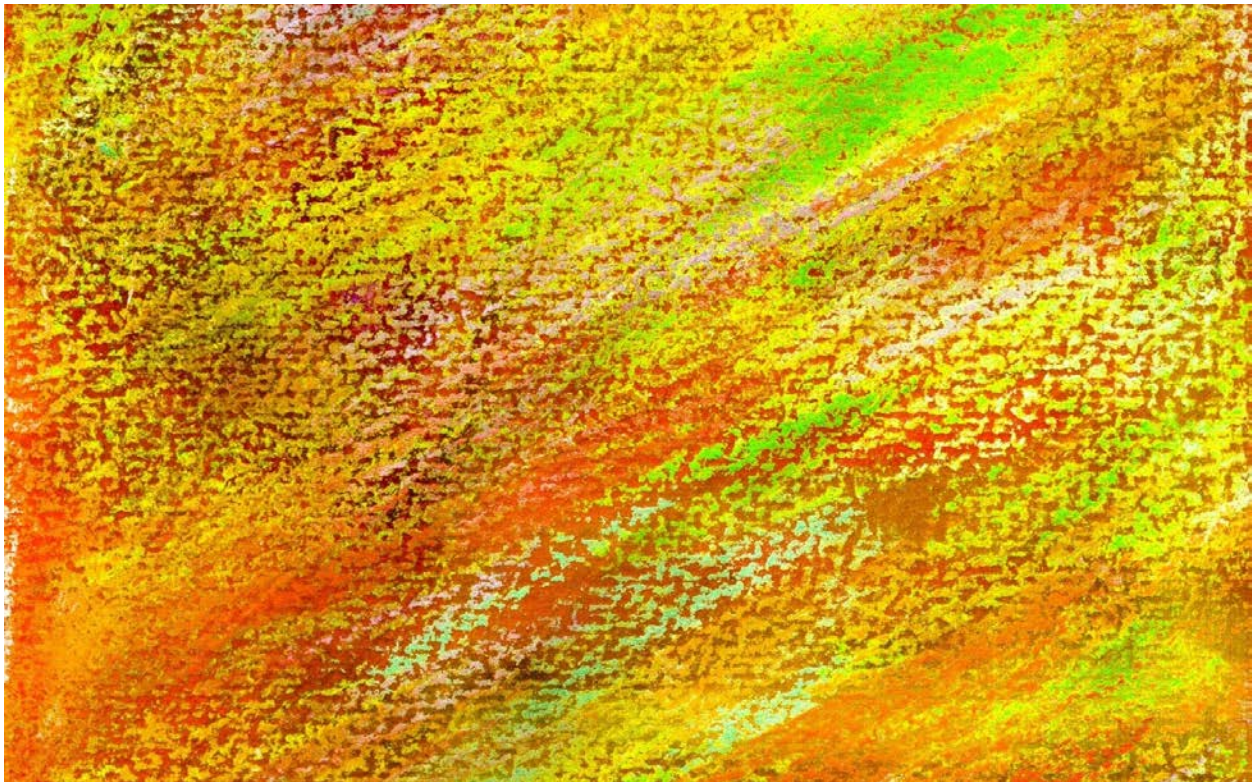
**David M. Kroenke • David J. Auer**

---

**Online Appendix C**

**Getting Started with  
MySQL 5.6 Community Server Edition**

---



---

**All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.**

---

**Editor in Chief:** Stephanie Wall  
**Acquisitions Editor:** Nicole Sam  
**Program Manager Team Lead:** Ashley Santora  
**Program Manager:** Denise Vaughn  
**Editorial Assistant:** Kaylee Rotella  
**Executive Marketing Manager:** Anne K. Fahlgren  
**Project Manager Team Lead:** Judy Leale  
**Project Manager:** Ilene Kahn  
**Procurement Specialist:** Michelle Klein

**Senior Art Director:** Janet Slowik  
**Text Designer:** Integra Software Services  
**Cover Designer:** Integra  
**Cover Art:** shibanuk/Fotolia  
**Full-Service Project Management:** Integra  
**Composition:** Integra  
**Printer/Binder:** Courier/Kendallville  
**Cover Printer:** Lehigh-Phoenix Color/Hagerstown  
**Text Font:** 10/12 Simoncini Garamond Std.

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on the appropriate page within text.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® Windows®, and Microsoft Office® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

MySQL®, the MySQL GUI Tools® (MySQL Query Browser® and MySQL Administrator®), the MySQL Command Line Client®, the MySQL Workbench®, and the MySQL Connector/ODBC® are registered trademarks of Sun Microsystems, Inc./Oracle Corporation. Screenshots and icons reprinted with permission of Oracle Corporation. This book is not sponsored or endorsed by or affiliated with Oracle Corporation.

Express Edition 11g Release 2 2014 by Oracle Corporation. Reprinted with permission.

PHP is copyright The PHP Group 1999–2012, and is used under the terms of the PHP Public License v3.01 available at [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt). This book is not sponsored or endorsed by or affiliated with The PHP Group.

---

Copyright © 2015, 2013, 2011 by Pearson Education, Inc., 221 River Street, Hoboken, New Jersey 07030. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright, and permission should 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. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 221 River Street, Hoboken, New Jersey 07030.

Many of the designations 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 in initial caps or all caps.

#### Library of Congress Cataloging-in-Publication Data

Kroenke, David M., 1948–  
 Database concepts / David M. Kroenke and David J. Auer.—7e.  
 pages cm  
 Includes index.  
 ISBN 978-0-13-354462-6 (student edition)—ISBN 978-0-13-354478-7 (instructor edition)  
 1. Database management. 2. Relational databases. I. Auer, David J. II. Title.  
 QA76.9.D3K736 2015  
 005.74—dc23

2014010915

## Objectives:

- Learn how to create a database in MySQL 5.5 Community Server Edition
- Learn how to submit SQL commands to create table structures
- Learn how to submit SQL commands to insert database data
- Learn how to submit SQL commands to query a database
- Learn how to install the MySQL ODBC/Connector
- Learn how to create database designs in the MySQL Workbench

## What is the Purpose of this Appendix?

**MySQL Community Server** is an open-source, freely downloadable, enterprise-class DBMS that has been around for many years. In November 2005, MySQL 5.0 was released, and, as of this writing, MySQL 5.6 is the generally available (GA) release, with MySQL 5.7 in development. In February 2008, Sun Microsystems completed its acquisition of MySQL AB, the company that created and owned MySQL. Subsequently, Oracle Corporation acquired Sun Microsystems in a deal that was finalized on January 27, 2010, after European Commission approval on January 21, 2010—see <http://www.sun.com/third-party/global/oracle/>). Thus, the Oracle Corporation now owns MySQL in addition to its flagship Oracle

Database product.<sup>1</sup> MySQL, while not having as many features as SQL Server, has become widely used and very popular as a DBMS supporting Web sites running the Apache Web server. The MySQL Community Server version and MySQL Workbench graphical user interface (GUI) utility are free.

### Why Should I Learn to Use MySQL?

For the purposes of this book, the most important reason to learn to use MySQL is that MySQL really handles SQL well. All the SQL commands and keywords in Chapter 3 and Appendix E marked “Does Not Work with Microsoft Access ANSI-89 SQL” will work with MySQL. There will still be minor variations in some SQL statements, but this is typical of DBMS products.

### What Will This Appendix Teach Me?

As its title implies, this appendix is designed to get you started creating databases and running SQL commands and **SQL scripts** (which are related groups of SQL commands) so that you can use a more robust SQL environment than provided by Microsoft Access.

### What Won't This Appendix Teach Me?

The material in this appendix does not go beyond the information necessary to get you started using MySQL. There are many important MySQL topics not covered here, including stored procedures, triggers, backups and restores, and database security. Some of these topics are covered in David M. Kroenke and David J. Auer, *Database Processing: Fundamentals, Design, and Implementation*, 13th edition (Upper Saddle River, NJ: Prentice Hall, 2014).

### How Do I Install MySQL 5.6 Community Server Edition?

MySQL runs on Linux, UNIX, NetWare, and even the Microsoft Windows operating systems. The various versions of Microsoft SQL Server (including the free SQL Server Express edition discussed in Appendix A), like all other Microsoft products, run only on Microsoft operating systems. If you are not using a Microsoft operating system, MySQL is an obvious contender for your DBMS of choice. MySQL is intended for general use and can be downloaded for free. You need to download at least the first two of the following three programs listed on the next page, but we recommend that you download all three.

---

<sup>1</sup> For information about Oracle Database Express Edition 11g Release 2, see online Appendix B—Getting Started with Oracle Database Express Edition 11g Release 2.

**IMPORTANT NOTE:** If you are using a **Windows operating system**, you should **only** install these programs by using the **MySQL Installer for Windows**, which can be downloaded from <http://dev.mysql.com/downloads/installer>. Do *not* try to install the programs separately on a Windows OS computer! If you do so, you may find that some functionality or connectivity between the programs is missing!

1. **MySQL Community Server Edition.**<sup>2</sup> Download the most current generally available version for your operating system. As this appendix was written, MySQL 5.6 was generally available, MySQL 5.7 was in development. Therefore, this appendix is based on MySQL 5.6 Community Server edition.
2. **MySQL Workbench.**<sup>3</sup> The MySQL Workbench has replaced the MySQL GUI Tools (The MySQL Query Browser and MySQL Administrator) as the graphical administration and SQL command utility for MySQL.<sup>4</sup> In addition, it can be used for creating database designs as described in Chapter 5.
3. **MySQL Connector/ODBC.** This provides the ODBC programs necessary to provide Web application connectivity to a MySQL database as described in Chapter 7.

### By The Way

If you look at the various Web sites for MySQL, you will find that the name of the **MySQL Community Server** edition varies from place to place. On the main Oracle MySQL Web site (see: <http://www.mysql.com/products>), it is referred to as *MySQL Community Edition*. On the MySQL Web site that we use to download the MySQL products (see: <http://dev.mysql.com/downloads/mysql>), it is referred to as both *MySQL Community Edition* and *MySQL Community Server*. On the MySQL Web site that we use to download the MySQL Installer for Windows (see: <http://dev.mysql.com/downloads/installer>), it is referred to as just the *MySQL Server*.

So, what shall we call it? In Database Concepts 7<sup>th</sup> Edition and all the accompanying online appendices, we use the term **MySQL Community Server**.

---

<sup>2</sup> This appendix, which was finalized after Database Concepts (7<sup>th</sup> Edition) itself went to press, uses MySQL 5.6 Community Server version 5.6.19. The screenshots in Database Concepts itself are based on MySQL 5.6 Community Server version 5.6.17. You may notice some slight variations in the screen shots from the two versions.

<sup>3</sup> This appendix, which was finalized after Database Concepts (7<sup>th</sup> Edition) itself went to press, uses MySQL Workbench 6.1.40 CE. The screenshots in Database Concepts Chapter 6 are based on MySQL Workbench 6.1.6 CE. You may notice some slight variations in the screen shots from the two versions.

<sup>4</sup> The MySQL Server GUI Tools reached “end of life” (EOL) on December 18, 2009. See the announcement at: <http://dev.mysql.com/support/eol-notice.html>. These tools are still available at <http://downloads.mysql.com/archives/gui/>. Previous editions of this book used the GUI Tools as the utilities for working MySQL.

MySQL and its associated utilities are very easy to install. At this point, install and configure MySQL 5.6 Community Server and the MySQL workbench by (1) using the MySQL Installer for Windows if you are using a Microsoft Windows OS or (2) downloading and installing each separate programs for non-Windows OSs. Install MySQL 5.6 Community Server as a *Developer Machine* when asked during the installation process. We will discuss the MySQL Connector/ODBC later in this appendix.

Note that during the installation and configuration of MySQL Community Server, you will be asked to provide a password for the *root* user account. **Root** is the name of the default MySQL administrator account. Be sure you provide the password when prompted, and be sure to remember this password! The username *Root* comes from the UNIX and Linux operating systems, where it is the name of the default system administrator account.

Be aware that MySQL 5.6 is an enterprise-class DBMS, and it is therefore much more complex than Microsoft Access. Further, it does not include application development tools, such as form and report generators.

### How Do I Create a Workspace for the MySQL Workbench Files?

Before using the MySQL Workbench we recommend creating a folder named **MySQL Workbench** under the **My Documents** folder (or whatever your main data storage area is named). In Windows, this can be done using Windows Explorer, as shown in Figure C-1. In this workspace, create a subfolder for each database project. In this workspace, create two folders, *EER Models* (for database designs) and *Schemas* (for database scripts). In the Schemas folder, create a subfolder for each database project each time you start a new database project.

### How Do I Start the MySQL Workbench?

To start the MySQL Workbench running in Microsoft Windows 8.1, click the **MySQL Workbench 6.1 CE** icon.<sup>5</sup> The MySQL Workbench splash screen is displayed, followed by the MySQL Workbench window with the Home page displayed, as shown in Figure C-2.

The MySQL Workbench Home tab is a dashboard allowing us access to MySQL database design (mis-labeled as “Models”), SQL database development and MySQL DBMS administration. There are several means of accessing the features provided, and we will only cover some basic ones here.<sup>6</sup>

---

<sup>5</sup> Alternatively, we recommend that you pin MySQL Workbench 6.1 CE to the Taskbar, as shown in Figures C-1 and C-2, and start the MySQL Workbench from this icon. The *CE* designation shows that this is the open source, freely downloadable community edition—there is also a commercial version available that includes additional functionality.

<sup>6</sup> If any models are listed in the Models section, right-click each model name and click Remove Model File from List.

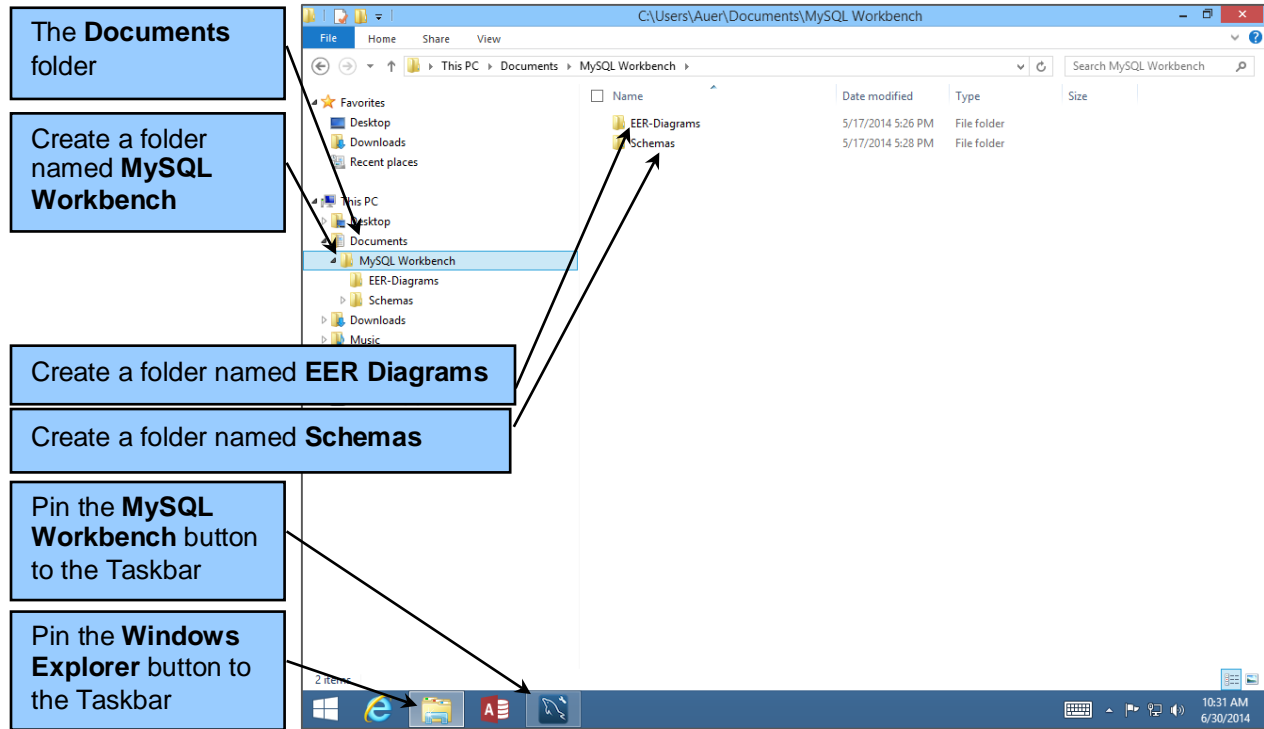


Figure C-1 — The MySQL Workbench Folder in Windows Explorer

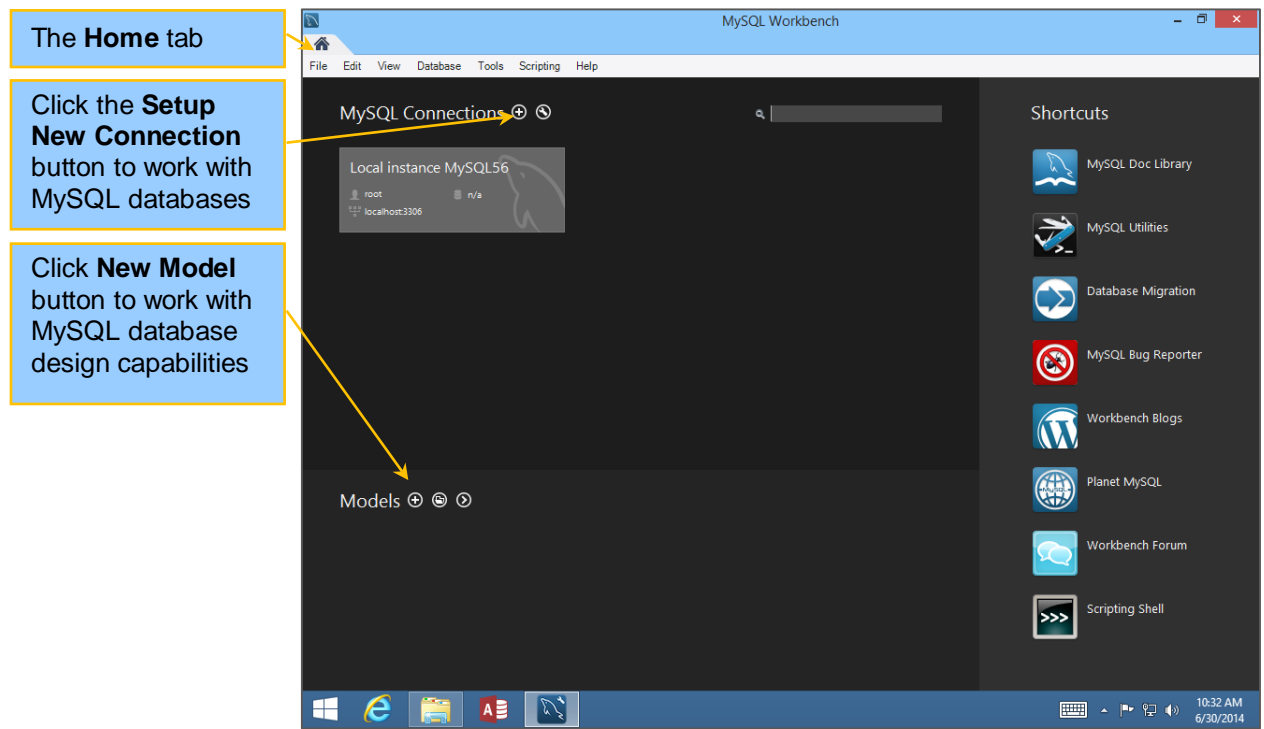


Figure C-2 — The MySQL Workbench Home Tab

## How Do I Create a Database in MySQL?

To create a database in MySQL Workbench, start by clicking the **MySQL Setup New Connection** button (the **Plus** button) in the **SQL Connections** section of the MySQL Workbench Home page (Figure C-2). The **Connect to Database** dialog box appears, as shown in Figure C-3. Enter the Username `root`<sup>7</sup> (if needed—it may be displayed by default), and then click the **OK** button. The **Connect to MySQL Server** dialog box is displayed, as shown in Figure C-4. Enter the password for the Root user, and then click the **OK** button. The MySQL Workbench reappears, but now with an open tabbed connection window named Local Instance MySQL56, as shown in Figure C-5.

### By The Way

Alternatively, if you click the **Local instance MySQL56** icon in the **SQL Development** section of the MySQL Workbench Home page, the **Connect to MySQL Sever** dialog box is immediately displayed.

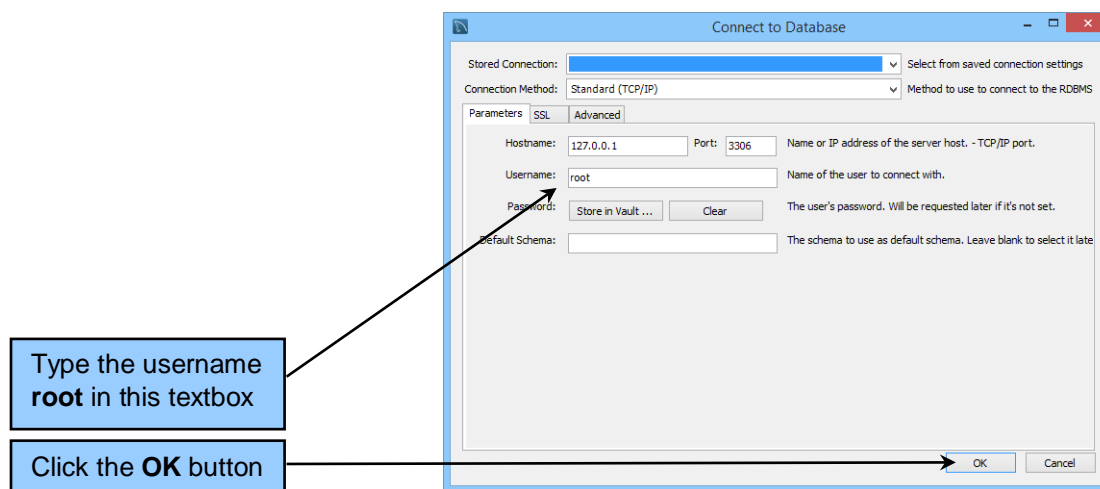


Figure C-3 — The Connect to Database Dialog Box

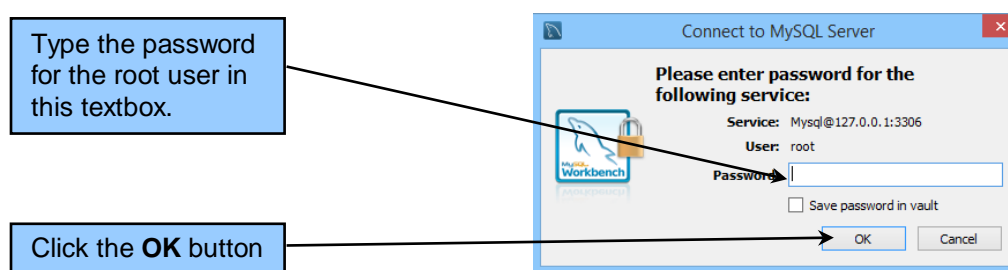


Figure C-4 — The Connect to MySQL Server Dialog Box

<sup>7</sup> As discussed earlier in this appendix, `root` is the name of the default MySQL administrator account. You are prompted for a password for this account during the installation of MySQL, and should have already created that password.



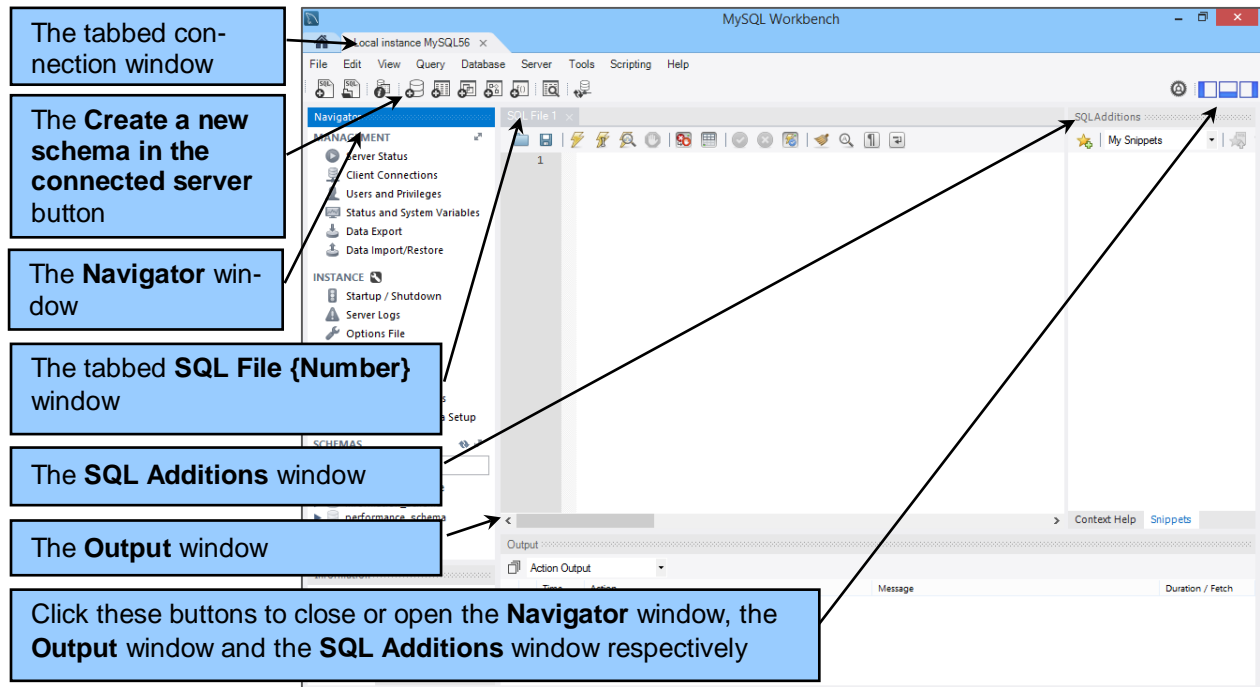


Figure C-5-A — The Navigator and the SQL File Tabbed Windows

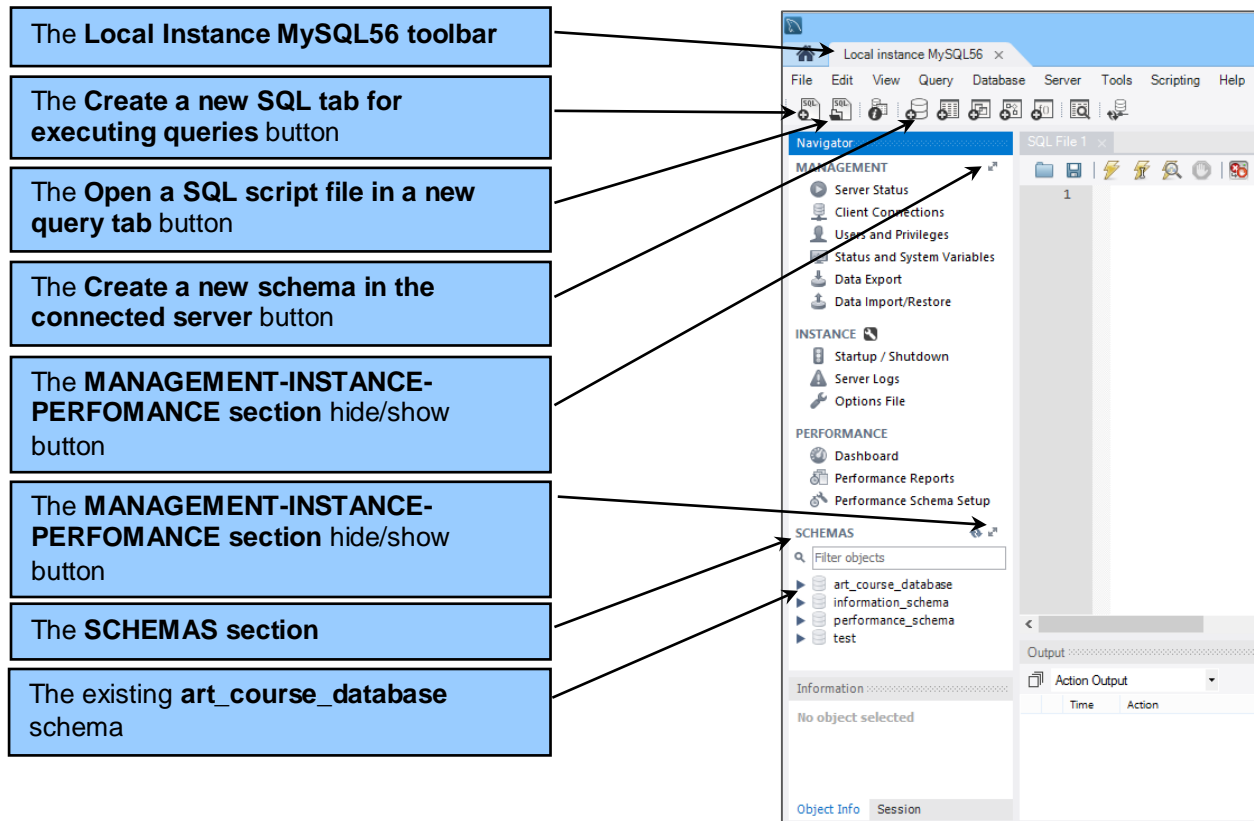


Figure C-5-B — The Navigator Window and Toolbars

In Figure C-5-A, we can see some of the features of the tabbed connection window:

1. The **Navigator**, which displays management functions and existing schemas—**Schema** is MySQL's term for a database. Note the existing *test* schema (automatically created by MySQL) and the *art\_course\_database* schema (that we used in Chapters 1 and 2).
2. The tabbed **SQL File {number}** window—We will use this as our SQL text editor.
3. The **Output** window—Test messages will be displayed here as we work on databases.
4. The **SQL Additions** window—We can store SQL code snippets (short bits of SQL code) here for reuse.
5. The three buttons that control the display of the Object Browser, the Output window and the SQL Additions window.
6. The **Create a new schema in the connected server** button—Yes, it is really named that, and it is used to create a new database (schema).

In Figure C-5-B, we can see some features of the Navigator window:

1. The **MANAGEMENT-INSTANCE-PERFORMANCE section**, which provide the management functions for the schemas.
2. The **MANAGEMENT-INSTANCE-PERFORMANCE section** hide/show buttons—Both will hide the management functions, and the button in the SCHEMAS section is used to show the functions if they are hidden.
3. The **Create a new SQL tab for executing queries** button—Self-explanatory, isn't it?
4. The **Open a SQL script file in a new query tab** button—Self-explanatory, isn't it?
5. The **Open a new schema in the connected server** button—Self-explanatory, isn't it?
6. The **SCHEMAS section**—This section is where we have control of the MySQL schemas.
7. The *art\_course\_database* schema—This is the database we worked with in Chapters 1 and 2.

We will not use the MANGEMENT-INSTANCE-PERFORMANCE section functionality nor the SQL Additions window shown in Figure C-5-A, so we will remove them from the display.

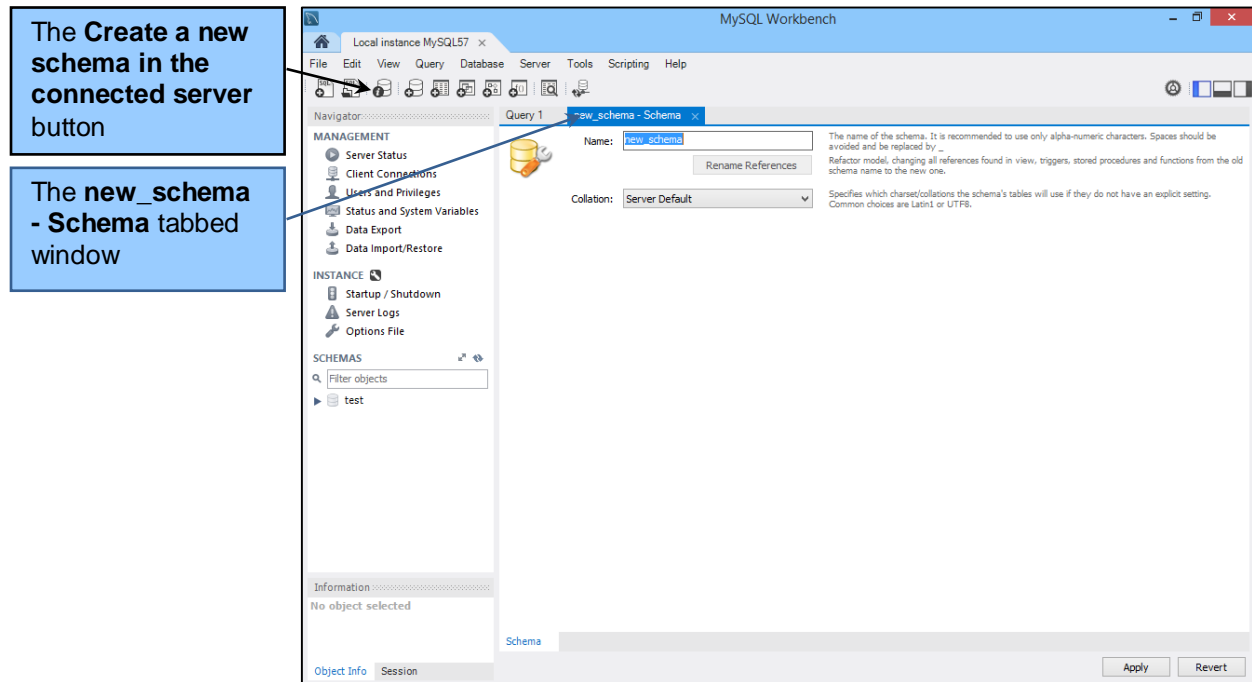
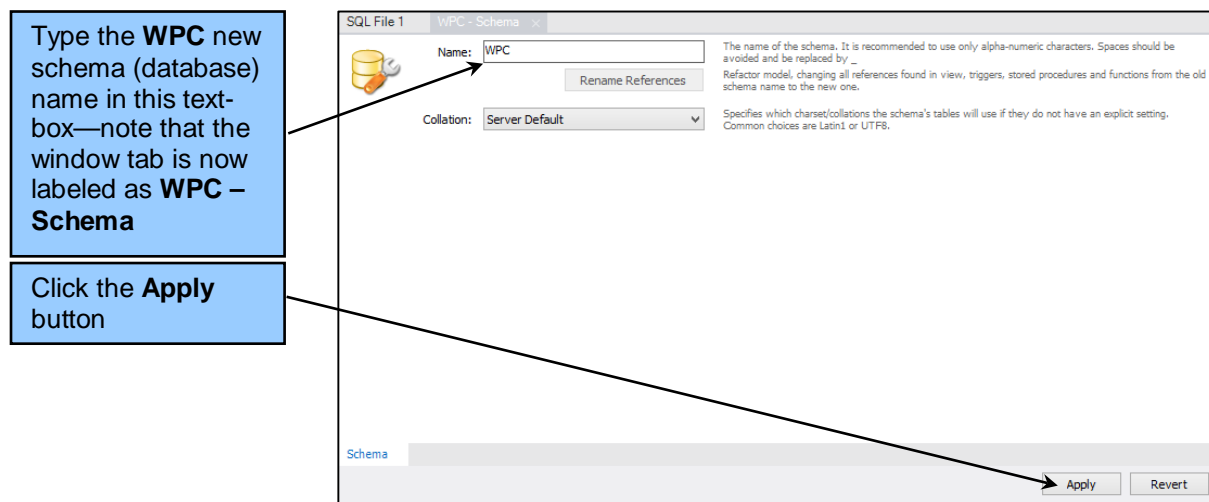
#### ***Controlling the MySQL Workbench SQL Editor Display:***

1. Click either of the MANGEMENT-INSTANCE\_PERFORMACE hide/show buttons (see the callouts in Figure C-5-B). The MANGEMENT-INSTANCE\_PERFORMACE section is no longer displayed, and the SCHEMA section expands into this display area.
2. Click the button that controls the SQL Additions window (see the last callout in Figure C-5-A). The SQL Additions window is no longer displayed, and the SQL File window expands into the display area previously used by the SQL Additions window.

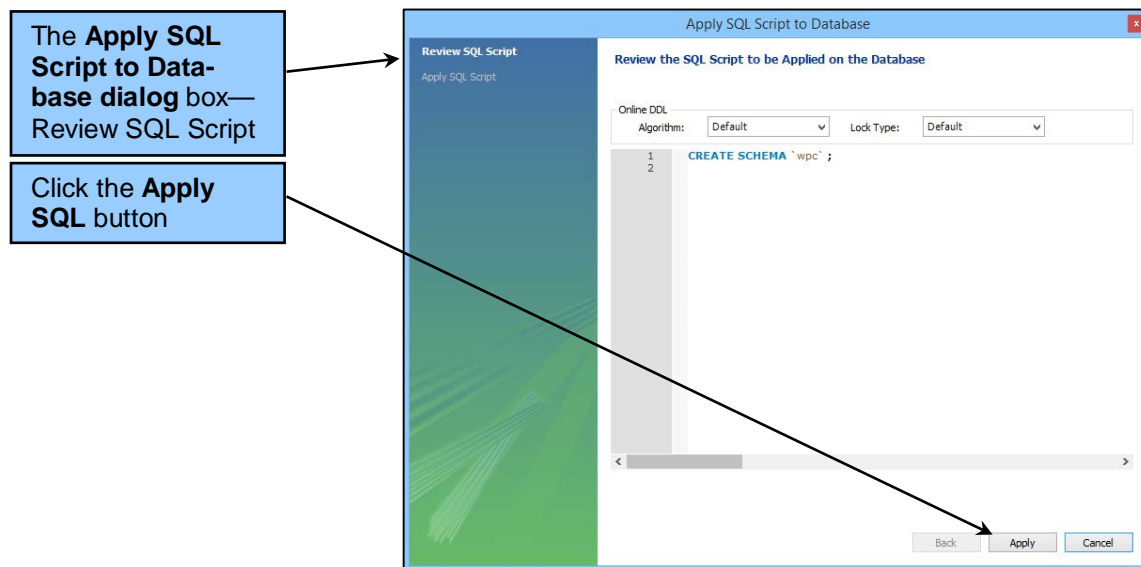
Now we will create the Wedgewood Pacific Corporation database that we used as our example database in Chapter 3 and Appendix E.

**Creating a MySQL database:**

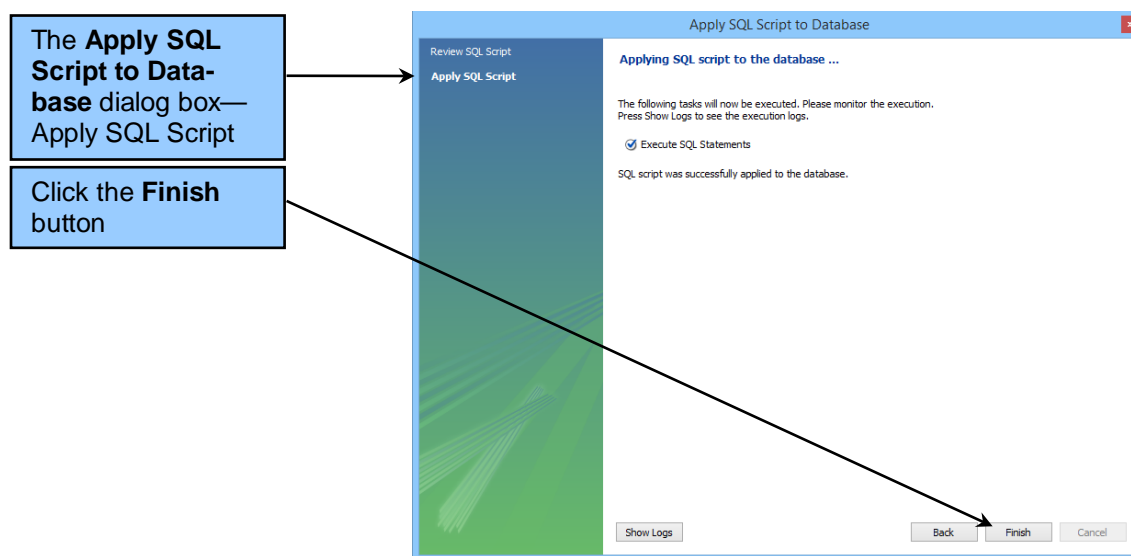
1. Click the **Create a new schema in the connected server** button shown in Figure C-5-B.
2. The new\_schema tabbed window is displayed, as shown in Figure C-6.
3. Type the new schema (database) name WPC in the **Name** textbox, as shown in Figure C-7, and then click the **Apply** button. A warning dialog box will declare that the schema name will be in lower case letters as “wpc.”

**Figure C-6 — The new\_schema Tabbed Window****Figure C-7 — The WPC Schema Name in the new\_schema Dialog Box**

- As shown in Figure C-8, an Apply SQL Script to Database dialog box is displayed so that the user can review the SQL command before it is executed. Click the **Apply** button.
- As shown in Figure C-9, the Apply SQL Script to Database dialog box is displayed with the results of executing the SQL command. Click the **Finish** button.
- The WPC (the renamed *new\_schema*) dialog box is displayed again. Click the **X (Close)** button.
- The WPC schema object now appears in the Object Browser. Note that although we entered the schema name as *WPC* (all upper case), MySQL insists on displaying it as *wpc* (all lower case). If you want to make sure the Object Browser objects are displayed correctly, then click **Refresh SCHEMAS Browser** button (see Figure C-10) which will refresh the display in the Object Browser.



**Figure C-8 — The Apply SQL Script to Database – Review the SQL Script to be Applied on the Database Dialog Box**



**Figure C-9 — The Apply SQL Script to Database – Applying SQL Script Dialog Box**

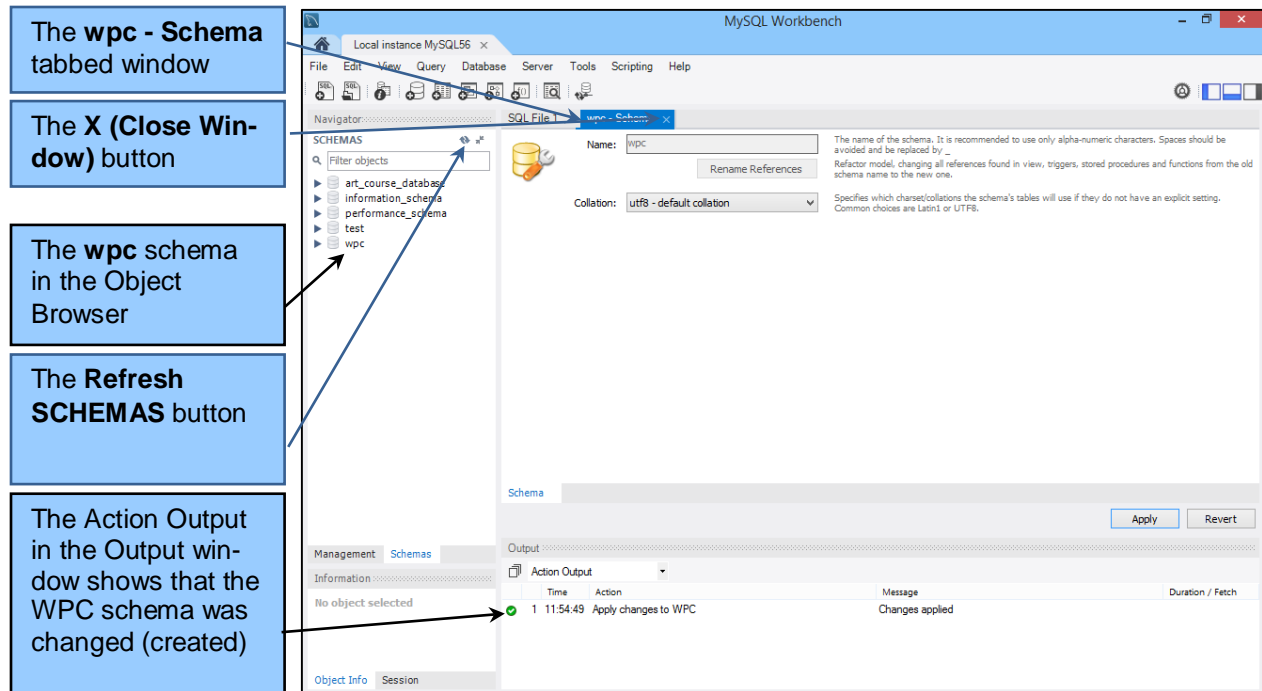


Figure C-10 — The wpc Schema (Database) in the Object Browser

## How Do I Set the Active Database in MySQL?

To work with a MySQL database, you must select it as the active database. In MySQL terms, this is called the **default schema**. To set the default schema, select the name of the database that you want to work with in the Default Schema drop-down list.

### Setting the Default Schema:

1. Click the **wpc** schema object in the object browser to select it.
2. Right-click the **wpc** schema object to display the shortcut menu, as shown in Figure C-11.
3. In the shortcut menu, click the **Set as Default Schema** command.
4. The wpc schema is set as the active schema (default schema), and the wpc schema object is displayed in bold text in the Object Browser to indicate that it is the active schema, as shown in Figure C-12.

## How Do I Work with SQL Statements in MySQL?

In MySQL, SQL statements can be run individually or as part of a related group of SQL statements known as an **SQL script**. Scripts, which are named with the **\*.sql** extension, are efficient for processing groups of SQL statements. For example, you could create a set of CREATE TABLE commands to build a new database structure as a script, or you could create a set of INSERT commands to use when data need to be added to a table as a script.

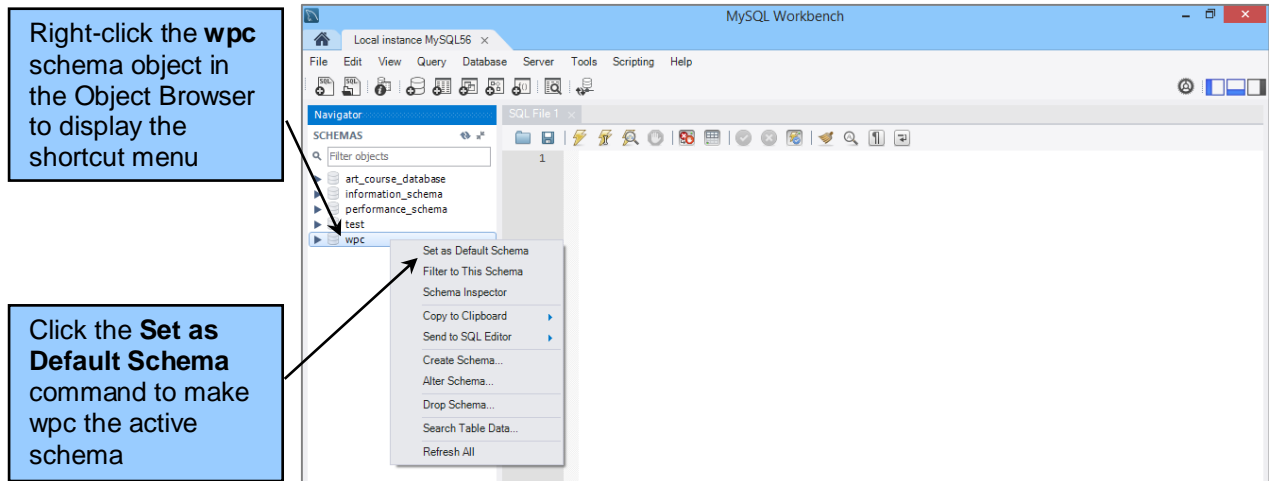


Figure C-11 — The Set as Default Schema Command

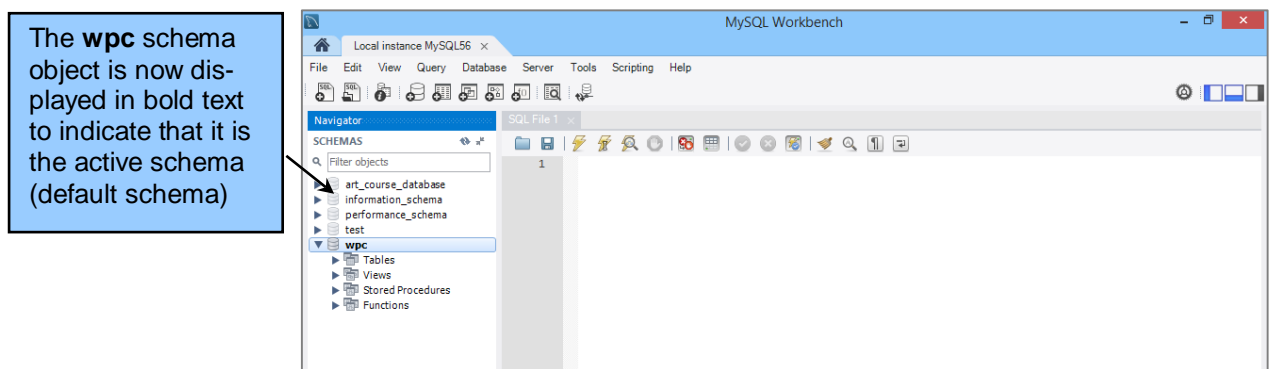


Figure C-12 — The wpc Schema as the Active Schema (Default Schema)

## SQL Commands to Create Table Structures

The SQL File tabbed window that is already open in MySQL Workbench is a script editor, and, in fact, is the preferred script editor when we are working with MySQL databases. Figure C-13 shows the MySQL version of the SQL statements to create the WPC database shown in Figure 3-7 [and MySQL data types are shown in Chapter 3 in Figure 3-5(c)]. Comparing the two versions of these SQL statements will help you understand any differences in how SQL is used in MySQL. The same statements are shown in MySQL Workbench in Figure C-14, together with some added header comments to document the script, saved as *DBC-e07-MySQL-WPC-Create-Tables.sql*.

## By The Way

MySQL uses the **AUTO\_INCREMENT** keyword to set surrogate key values. The default values for **AUTO\_INCREMENT** are an initial value of 1 and an increment of 1. After the table is created, you can modify the initial value by using the **ALTER TABLE** statement. For example, in the **WPC** database, the **PROJECT** table has a surrogate key that starts at 1000 and increments by 100. You could set the correct initial value by using the SQL statement:

```
ALTER TABLE PROJECT AUTO_INCREMENT=1000;
```

Unfortunately, you cannot change the MySQL increment of 1. So for cases such as the **ProjectID** values in **PROJECT**, you need to manually insert the correct values in MySQL.

```

/*****
/*
/* Kroenke and Auer - Database Concepts (7th Edition) Chapter 03
/*
/* Wedgewood Pacific Corporation [WPC] Create Tables
/*
/* These are the MySQL 5.6 SQL code solutions
/*
/* NOTE: ALTER TABLE statements can be used to set the
/* MySQL AUTO_INCREMENT to a starting value other than 1. For example:
/*
/* ALTER TABLE PROJECT AUTO_INCREMENT=1000;
/*
/* However, the increment is always one (1), so PROJECT ProjectID
/* values must be manually inserted
/*
/*
/*****

CREATE TABLE DEPARTMENT(
    DepartmentName Char(35) NOT NULL,
    BudgetCode Char(30) NOT NULL,
    OfficeNumber Char(15) NOT NULL,
    Phone Char(12) NOT NULL,
    CONSTRAINT DEPARTMENT_PK PRIMARY KEY(DepartmentName)
);

CREATE TABLE EMPLOYEE(
    EmployeeNumber Int NOT NULL AUTO_INCREMENT,
    FirstName Char(25) NOT NULL,
    LastName Char(25) NOT NULL,
    Department Char(35) NOT NULL DEFAULT 'Human Resources',
    Phone Char(12) NULL,
    Email VarChar(100) NOT NULL UNIQUE,
    CONSTRAINT EMPLOYEE_PK PRIMARY KEY(EmployeeNumber),
    CONSTRAINT EMP_DEPART_FK FOREIGN KEY (Department)
        REFERENCES DEPARTMENT(DepartmentName)
        ON UPDATE CASCADE
);

```

**Figure C-13 — The WPC Database Table SQL Statements for MySQL**

```

CREATE TABLE PROJECT(
    ProjectID          Int          NOT NULL,
    ProjectName       Char(50)     NOT NULL,
    Department        Char(35)     NOT NULL,
    MaxHours          Decimal(8,2) NOT NULL DEFAULT 100,
    StartDate        Date          NULL,
    EndDate           Date          NULL,
    CONSTRAINT        PROJECT_PK   PRIMARY KEY (ProjectID),
    CONSTRAINT        PROJ_DEPART_FK FOREIGN KEY (Department)
                                REFERENCES DEPARTMENT(DepartmentName)
                                ON UPDATE CASCADE
);

CREATE TABLE ASSIGNMENT (
    ProjectID          Int          NOT NULL,
    EmployeeNumber     Int          NOT NULL,
    HoursWorked       Numeric(6,2) NULL,
    CONSTRAINT        ASSIGNMENT_PK PRIMARY KEY(ProjectID, EmployeeNumber),
    CONSTRAINT        ASSIGN_PROJ_FK FOREIGN KEY (ProjectID)
                                REFERENCES PROJECT (ProjectID)
                                ON UPDATE NO ACTION
                                ON DELETE CASCADE,
    CONSTRAINT        ASSIGN_EMP_FK FOREIGN KEY (EmployeeNumber)
                                REFERENCES EMPLOYEE(EmployeeNumber)
                                ON UPDATE NO ACTION
                                ON DELETE NO ACTION
);

/*****

```

Figure C-13 — The WPC Database Table SQL Statements for MySQL (continued)

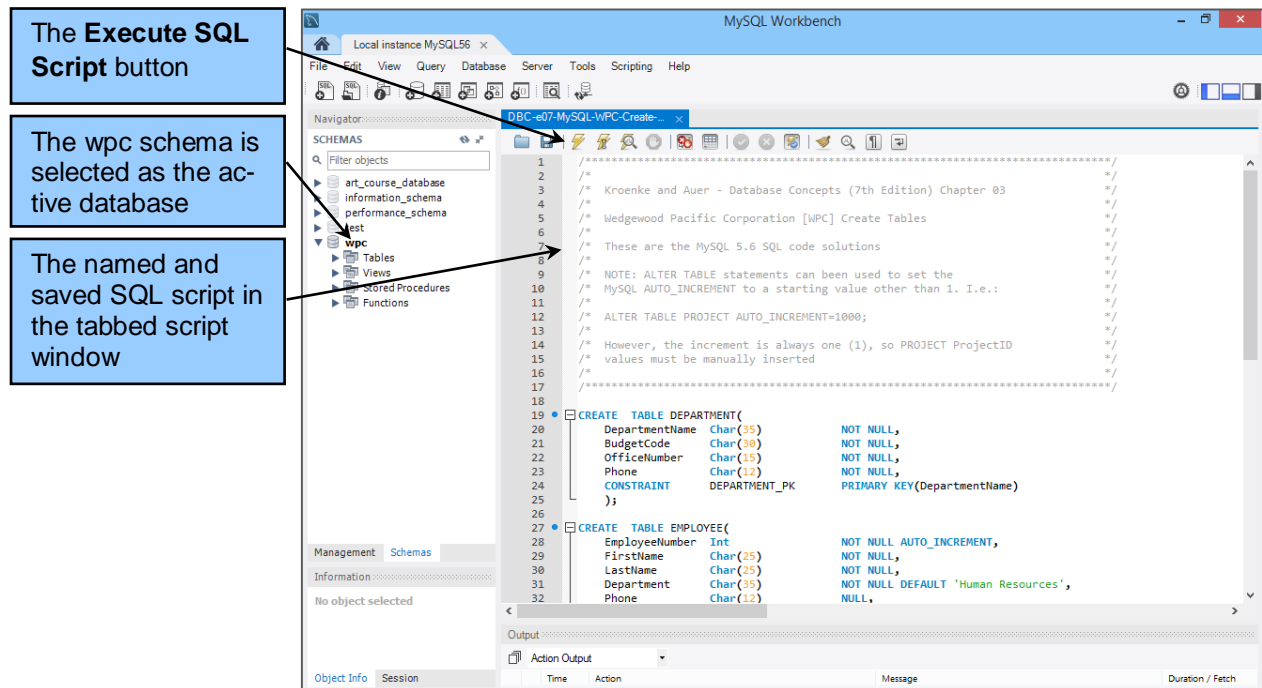


Figure C-14 — The WPC Script in the MySQL Workbench



## How Do I Execute an SQL Script in MySQL Workbench?

After the script has been named and saved (it is named during the save), make sure that WPC is the active database, and then click the **Execute SQL Script** button (the *thunderbolt* icon) to run the script. After the script executes, you can expand the WPC schema object to display the tables created by the script. To close the Script window, click the **X** symbol on the Script window tab. The results are shown in Figure C-15.

### By The Way

Believe it or not, the full name of the button that we have labeled as the *Execute SQL Script* button is the **Execute the selected portion of the script or everything, if there is no selection SQL** button according to the mouse roll-over tool-tip. To see this for yourself, move the mouse over the button and leave it there for a few seconds. A tool tip window will open with the full name of the button. The full name is very descriptive and accurate, but a bit long to be used in our instructions!

## SQL Commands to Insert Data

Figure C-16 shows the MySQL version of the SQL statements to populate the WPC database with data shown in Figure 3-12. Comparing the two versions of these SQL statements will help you understand any differences in how SQL is used in MySQL. Create and save an SQL script named *DBC-e07-MySQL-WPC-Insert-Data.sql* based on Figure C-16, and then execute the script to populate the WPC database.

The screenshot displays the MySQL Workbench interface. The Navigator on the left shows the 'wpc' schema expanded under 'Tables'. The central SQL editor contains the following SQL commands:

```

19 CREATE TABLE DEPARTMENT(
20   DepartmentName Char(35) NOT NULL,
21   BudgetCode Char(30) NOT NULL,
22   OfficeNumber Char(15) NOT NULL,
23   Phone Char(12) NOT NULL,
24   CONSTRAINT DEPARTMENT_PK PRIMARY KEY(DepartmentName)
25 );
26
27 CREATE TABLE EMPLOYEE(
28   EmployeeNumber Int NOT NULL AUTO_INCREMENT,
29   FirstName Char(25) NOT NULL,
30   LastName Char(25) NOT NULL,
31   Department Char(35) NOT NULL DEFAULT 'Human Resources',
32   Phone Char(12) NULL,
33   Email VarChar(100) NOT NULL UNIQUE,
34   CONSTRAINT EMPLOYEE_PK PRIMARY KEY(EmployeeNumber),
35   CONSTRAINT EMP_DEPART_FK FOREIGN KEY (Department)
36   REFERENCES DEPARTMENT(DepartmentName)
37   ON UPDATE CASCADE
38 );
39
40 CREATE TABLE PROJECT(
41   ProjectID Int NOT NULL,
42   ProjectName Char(50) NOT NULL,
43   Department Char(35) NOT NULL,
44   ...

```

The Output window at the bottom shows the 'Action Output' table with the following data:

Time	Action	Message	Duration / Fetch
2 18:03:38	CREATE TABLE DEPARTMENT( DepartmentNameChar(...	0 row(s) affected	0.672 sec
3 18:03:39	CREATE TABLE EMPLOYEE( EmployeeNumberInt ...	0 row(s) affected	0.203 sec
4 18:03:39	CREATE TABLE PROJECT( ProjectID Int NOT NU...	0 row(s) affected	0.172 sec
5 18:03:40	CREATE TABLE ASSIGNMENT( ProjectID Int N...	0 row(s) affected	0.187 sec

Figure C-15 — The SQL Script Results

```

/*****
/*
/* Kroenke and Auer - Database Concepts (7th Edition) Chapter 03
/*
/* Wedgewood Pacific Corporation [WPC] Database Data
/*
/* These are the MySQL 5.6 SQL code solutions
/*
/*
/*****/

/***** DEPARTMENT DATA *****/
INSERT INTO DEPARTMENT
VALUES('Administration', 'BC-100-10', 'BLDG01-300', '360-285-8100');
INSERT INTO DEPARTMENT
VALUES('Legal', 'BC-200-10', 'BLDG01-200', '360-285-8200');
INSERT INTO DEPARTMENT
VALUES('Accounting', 'BC-300-10', 'BLDG01-100', '360-285-8300');
INSERT INTO DEPARTMENT
VALUES('Finance', 'BC-400-10', 'BLDG01-140', '360-285-8400');
INSERT INTO DEPARTMENT
VALUES('Human Resources', 'BC-500-10', 'BLDG01-180', '360-285-8500');
INSERT INTO DEPARTMENT
VALUES('Production', 'BC-600-10', 'BLDG02-100', '360-287-8600');
INSERT INTO DEPARTMENT
VALUES('Marketing', 'BC-700-10', 'BLDG02-200', '360-287-8700');
INSERT INTO DEPARTMENT
VALUES('InfoSystems', 'BC-800-10', 'BLDG02-270', '360-287-8800');

/***** EMPLOYEE DATA *****/

INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Mary', 'Jacobs', 'Administration', '360-285-8110', 'Mary.Jacobs@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Rosalie', 'Jackson', 'Administration', '360-285-8120', 'Rosalie.Jack-
son@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Richard', 'Bandalone', 'Legal', '360-285-8210', 'Richard.Banalone@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Tom', 'Caruthers', 'Accounting', '360-285-8310', 'Tom.Caruthers@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Heather', 'Jones', 'Accounting', '360-285-8320', 'Heather.Jones@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Mary', 'Abernathy', 'Finance', '360-285-8410', 'Mary.Abernathy@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'George', 'Smith', 'Human Resources', '360-285-8510', 'GeorgeSmith@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'Tom', 'Jackson', 'Production', '360-287-8610', 'Tom.Jackson@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
'George', 'Jones', 'Production', '360-287-8620', 'George.Jones@WPC.com');

```

---

**Figure C-16 — The WPC Database Data INSERT Statements for MySQL**

```
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
  'Ken', 'Numoto', 'Marketing', '360-287-8710', 'Ken.Numoto@WPC.com');
INSERT INTO EMPLOYEE(FirstName, LastName, Department, Email)
VALUES(
  'James', 'Nestor', 'InfoSystems', 'James.Nestor@WPC.com');
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Phone, Email)
VALUES(
  'Rick', 'Brown', 'InfoSystems', '360-287-8820', 'Rick.Brown@WPC.com');

/*****   PROJECT DATA   *****/

INSERT INTO PROJECT VALUES(
  1000, '2014 Q3 Product Plan', 'Marketing', 135.00, '2014-05-10', '2014-06-15');
INSERT INTO PROJECT VALUES(
  1100, '2014 Q3 Portfolio Analysis', 'Finance', 120.00, '2014-07-05', '2014-07-25');
INSERT INTO PROJECT VALUES(
  1200, '2014 Q3 Tax Preparation', 'Accounting', 145.00, '2014-08-10', '2014-10-15');
INSERT INTO PROJECT VALUES(
  1300, '2014 Q4 Product Plan', 'Marketing', 150.00, '2014-08-10', '2014-09-15');
INSERT INTO PROJECT (ProjectID, ProjectName, Department, MaxHours, StartDate)
VALUES(
  1400, '2014 Q4 Portfolio Analysis', 'Finance', 140.00, '2012-10-05');

/*****   ASSIGNMENT DATA   *****/

INSERT INTO ASSIGNMENT VALUES(1000, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1000, 8, 75.0);
INSERT INTO ASSIGNMENT VALUES(1000, 10, 55.0);
INSERT INTO ASSIGNMENT VALUES(1100, 4, 40.0);
INSERT INTO ASSIGNMENT VALUES(1100, 6, 45.0);
INSERT INTO ASSIGNMENT VALUES(1200, 1, 25.0);
INSERT INTO ASSIGNMENT VALUES(1200, 2, 20.0);
INSERT INTO ASSIGNMENT VALUES(1200, 4, 45.0);
INSERT INTO ASSIGNMENT VALUES(1200, 5, 40.0);
INSERT INTO ASSIGNMENT VALUES(1300, 1, 35.0);
INSERT INTO ASSIGNMENT VALUES(1300, 8, 80.0);
INSERT INTO ASSIGNMENT VALUES(1300, 10, 50.0);
INSERT INTO ASSIGNMENT VALUES(1400, 4, 15.0);
INSERT INTO ASSIGNMENT VALUES(1400, 5, 10.0);
INSERT INTO ASSIGNMENT VALUES(1400, 6, 27.5);

/*****/
```

---

**Figure C-16 — The WPC Database Data INSERT Statements for MySQL (continued)**

---

## How Do I Work with SQL Queries in MySQL?

Now that we've created and populated the WPC database, we can run SQL queries against the data. While scripts are good for large sets of SQL commands that need to be run together, most SQL queries are run as single commands. Nonetheless, individual SQL queries are created in the same tabbed window, and run the same, as SQL scripts. However, to be consistent with the SQL conventions we use in this book, we need to configure how SQL keywords appear in the MySQL Workbench SQL Editor. Select **Edit | Preferences** and then click the **SQL Editor** tab to display the Workbench preferences shown in

C-17-A. Click the **Use UPPERCASE keywords on completion** check box to configure the SQL Editor to display SQL keywords in uppercase.

Having done this, Figure C-17-B shows an SQL query with its results. SQL queries can also be named and saved. As shown in Figure C-17-B, the results of the SQL query appear in a tabbed results window in a spreadsheet-style display. You can modify the column widths in the results display by using standard Windows drag-and-drop techniques to help make more data visible. You can run multiple queries at the same time by using multiple results windows. You can open additional SQL Script tabbed windows so that you can work with multiple SQL queries or SQL scripts at the same time. You can also highlight a single SQL command in a script and run it separately using the **Execute SQL Script** button shown in Figure C-17-B (the full name of the button is the *Execute the selected portion of the script or everything, if there is no selection* button, but we'll settle for the short version).

## Additional Documentation for MySQL

To get access to MySQL documentation, use the **Help | Help Index** command for help on MySQL Workbench itself, and go to <http://dev.mysql.com/doc> for documentation on MySQL 5.6 (and other versions of MySQL).

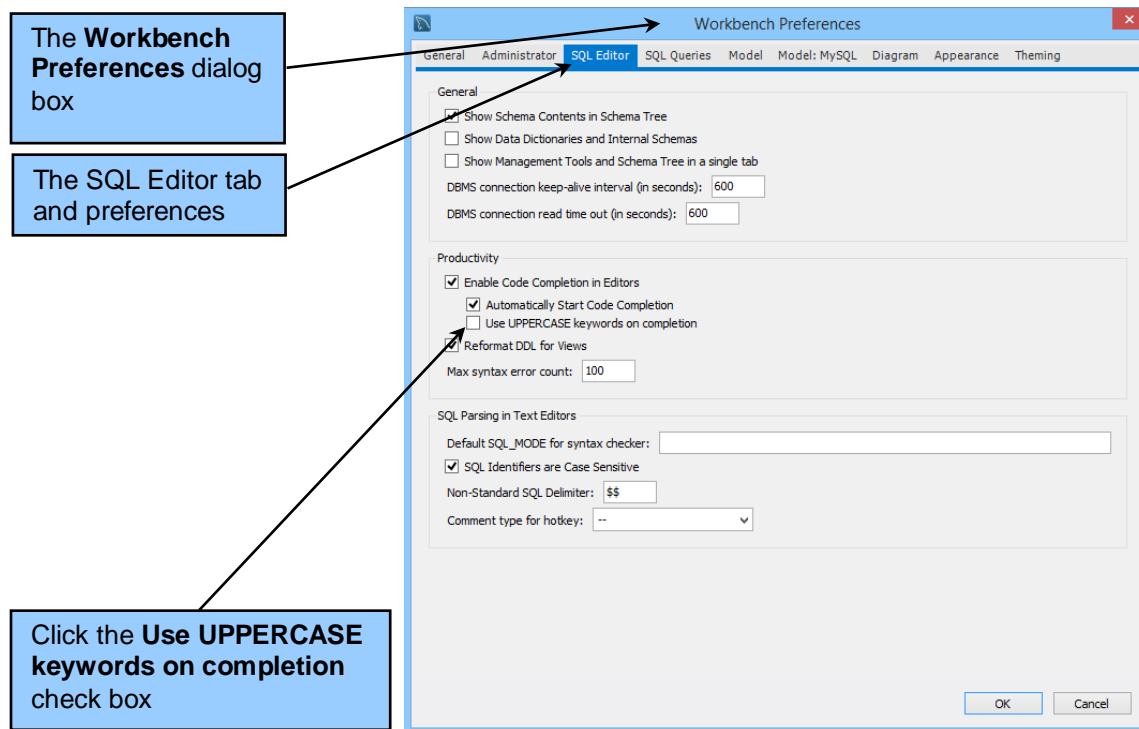


Figure C-17-A — The SQL Workbench SQL Editor Preferences

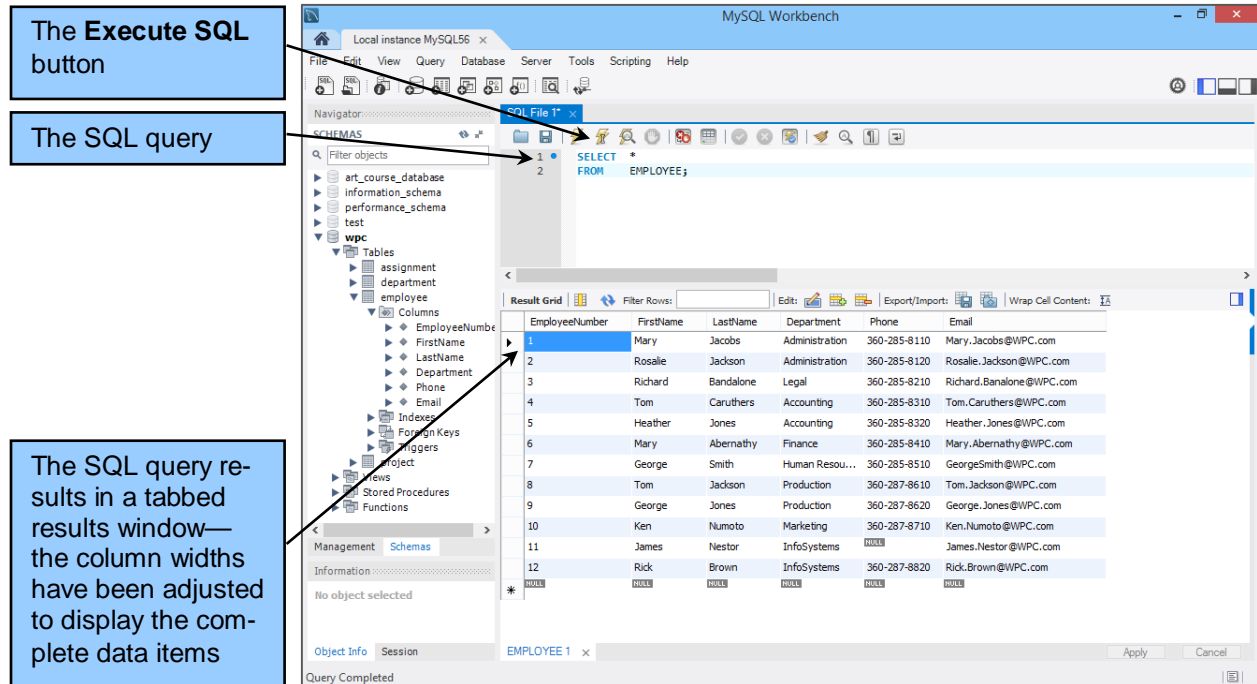


Figure C-17-B — The SQL Query and Results in the MySQL Workbench

### By The Way

The folks who work on the MySQL Workbench have **no qualms** about making **major GUI changes** in minor version upgrades of the product. The same functionality will be there, but perhaps implemented slightly differently! **Be aware and be prepared!**

### How Do I Install the MySQL Connector/ODBC?

We will use *open database connectivity (ODBC)* in Chapter 7 when we create Web pages that link to databases. Unlike the installation of Microsoft Access 2013 (separately or as part of Office 2013) or Microsoft SQL Server 2014, where the ODBC support is installed as part of the DBMS installation, you have to take an extra step to install ODBC support for MySQL. However, this simply requires downloading the MySQL Connector/ODBC from the MySQL Web site at <http://dev.mysql.com/downloads/connector/odbc/>, and running the installation routine. As soon as the installation is complete, the MySQL ODBC 5.3.4 drivers will be available in the ODBC Data Source Administrator.

**REMINDER:** If you are using a **Windows operating system**, you should **only** install the MySQL Connector/ODBC (and any of the other MySQL Connectors available from MySQL if you need them) by using the **MySQL Installer for Windows**, which can be downloaded from <http://dev.mysql.com/downloads/installer>. Do **not** try to install the MySQL Connector/ODBC separately on a Windows OS computer!

## How Do I Create Database Designs in the MySQL Workbench?

As we discuss in Chapters 4 and 5, a **data model** is a logical or conceptual view of the database. A **database design** defines the database characteristics that will be implemented in the actual database. One of the main differences between a data model and a database design is how N:M relationships are handled. In a data model, N:M relationships exist as N:M non-identifying relationships between two strong entities. In a database design, N:M relationships are broken into two 1:N identifying relationships between three ID-dependent entities.

While the MySQL Workbench refers to “data modeling” capabilities, these are really database design capabilities. The MySQL Workbench cannot create true N:M relationships, only the two 1:N relationships between three entities (the two original entities in the true data model and an intersection table).

We will illustrate creating database designs in the MySQL Workbench by creating an E-R diagram for the Wedgewood Pacific Corporation (WPC) database that we created earlier in this appendix.<sup>8</sup> When we are done, we will have a database diagram that will closely resemble Figures 5-16 and 5-17.

## How Do I Create a Database Model and E-R Diagram in the MySQL Workbench?

To create a new data model in the MySQL Workbench, you can:

- Use the **File | New Model** command, or
- Click the **New Model** button on the Home page.

Once the model is created, entity-relationship (E-R) diagrams, which the MySQL Workbench refers to as EER for the **extended entity-relationship** model (which, as discussed in Chapter 4, is the correct term, although today the term E-R model always means the EER model), are created within the model.

Note that you can create database designs *without* connecting to a MySQL server. In our case, we have already connected to the MySQL server because we created the WPC database.

---

<sup>8</sup> Of course, it could be argued that we really should have created the database design first, and then implemented that design. In many database courses, the data modeling and database design topics (which we cover in Chapter 4 and 5) are taught before using SQL to create the databases (which we cover in Chapter 3). In this case, the database design will follow the actual implementation of the database in the DBMS. We prefer to introduce SQL earlier. There are two reasons for this. First, users who are never involved in creating databases still often use SQL or QBE for querying databases (usually in data warehouses or data marts as discussed in Chapter 8) to gather information. Second, we like to get our students involved with DBMSs, databases and SQL as early in the course as possible. Either approach works and your Professor will choose the one that he or she prefers.

### Creating a New MySQL E-R (EER) Diagram:

1. If needed, open the MySQL Workbench. Since we haven't closed the MySQL Workbench in previous steps, we will assume that it is open with an open SQL Editor tabbed window displayed.
2. Click the **Home** page tab to return to the MySQL Workbench home page.
3. Click the **File | New Model** command. The MySQL Model tabbed window is displayed, as shown in Figure C-18.
4. In the MySQL Model tabbed window, click the **File | Save Model As...** command to display the **Save Model** dialog box. Browse to the Documents\MySQL Workbench\EER-Diagrams folder, and click the **New folder** button. Name the new folder **WPC-Database**.
5. In the WPC-Database folder, save the new model as **WPC-Database-Design.mwb** (note that *.mwb* is the default file extension used by the MySQL Workbench for data models).
6. Double click the **Add Diagram** icon in the Model Overview section of the MySQL Model page. A new, blank EER Diagram page is displayed, as shown in Figure C-19. Again note that MySQL uses the acronym *EER* where we are using *E-R*.
7. Click the **Save Model to Current File** button to save the WPC-Database-Design model with the E-R diagram.

Now that we have a blank E-R diagram work area available, we can build the E-R diagram itself. We start by adding a table to the E-R diagram. We will add the DEPARTMENT table. By looking at the SQL statement for this table in Figure C-13, we can see the columns that are used in the DEPARTMENT table.

Here is the SQL statement for the DEPARTMENT table:

```
CREATE TABLE DEPARTMENT(  
    DepartmentName    Char(35)          NOT NULL,  
    BudgetCode        Char(30)          NOT NULL,  
    OfficeNumber       Char(15)          NOT NULL,  
    Phone              Char(12)         NOT NULL,  
    CONSTRAINT        DEPARTMENT_PK    PRIMARY KEY(DepartmentName)  
);
```

### Creating a Table in the MySQL E-R (EER) Diagram:

1. In the E-R diagram toolbar, click the **Place a New Table** button shown in Figure C-19.
2. Move the cursor over the blank E-R diagram area. As you do so, notice that table property options are displayed across the top of the design area, as shown in Figure C-20. We do not need to change any of these.

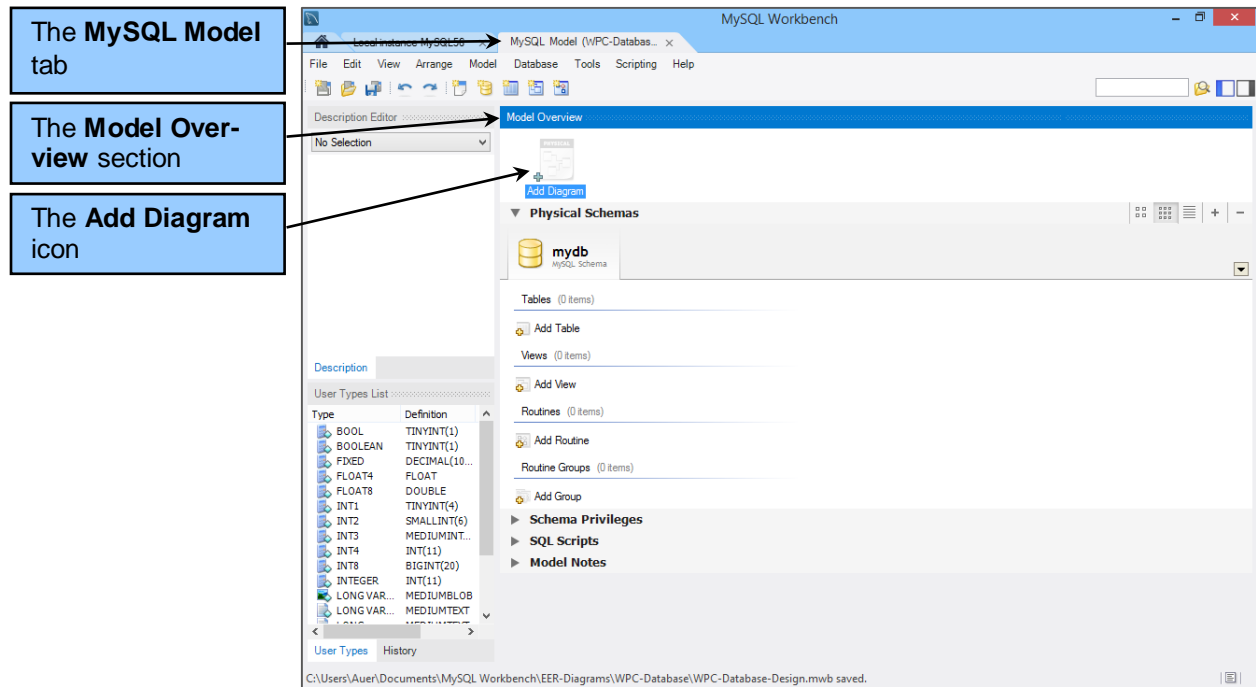


Figure C-18 — The MySQL Model Tab and Window

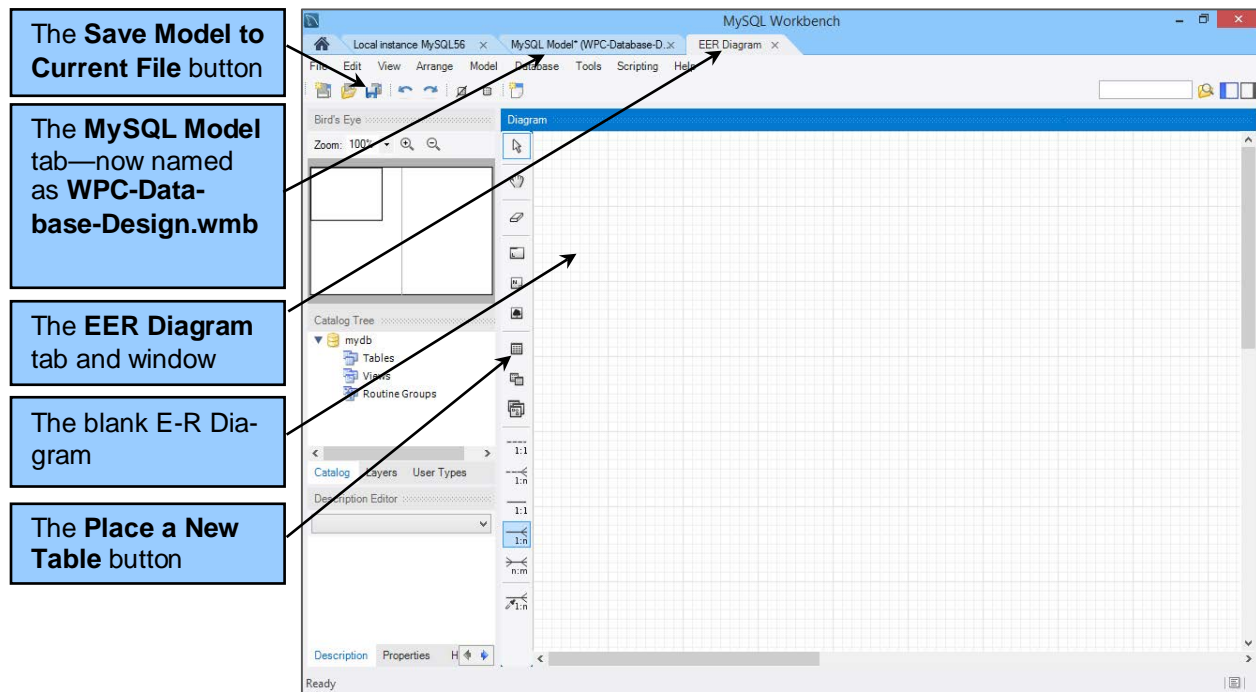


Figure C-19 — The Blank E-R Diagram



3. When the cursor is at the place where you want the table icon, click the left mouse button. A new table object named `table1` is created on the E-R Diagram, as shown in Figure C-21.
4. Double-click the **table1** object to open the MySQL table editor for `table1` as shown in Figure C-22. Note that the table editor has tabs along the bottom of the editor window, and that the Columns tab is selected so that the Columns page is displayed.
  - **NOTE:** Alternatively, right-click the table object to display a shortcut menu, and then click Edit Table... .
5. In the Columns page, click the **Table Name** text box to select it if needed, and type in the table name **DEPARTMENT**. Note that the new table name immediately appears on the table editor tab and the table object in the E-R diagram area, as shown in Figure C-23.
6. If necessary, drag the right boundary of the Columns area to the right so that all the column property settings are displayed. Figure C-23 shows the results of this operation, where now a check box labeled *AI* and a text box labeled *Default* can be seen.

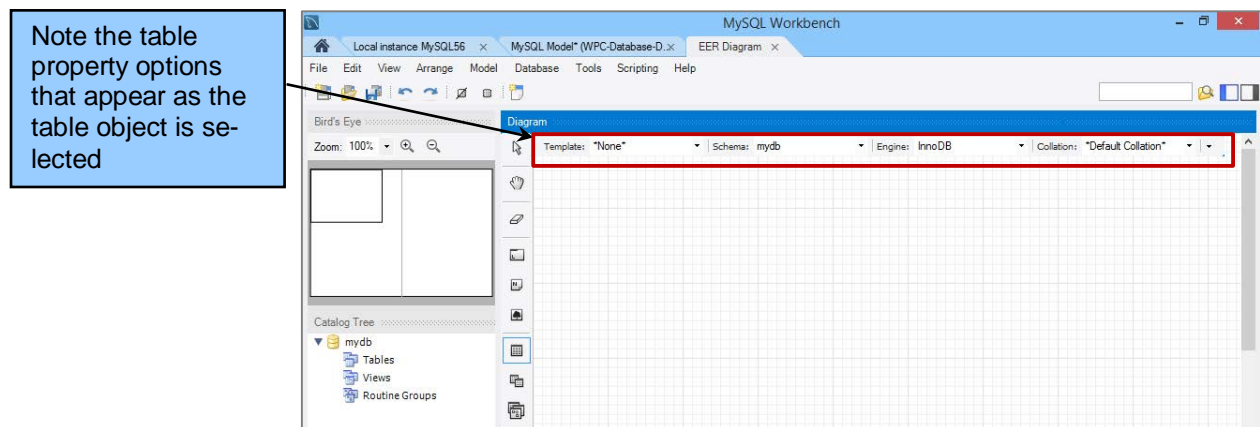


Figure C-20 — Placing the Table Object

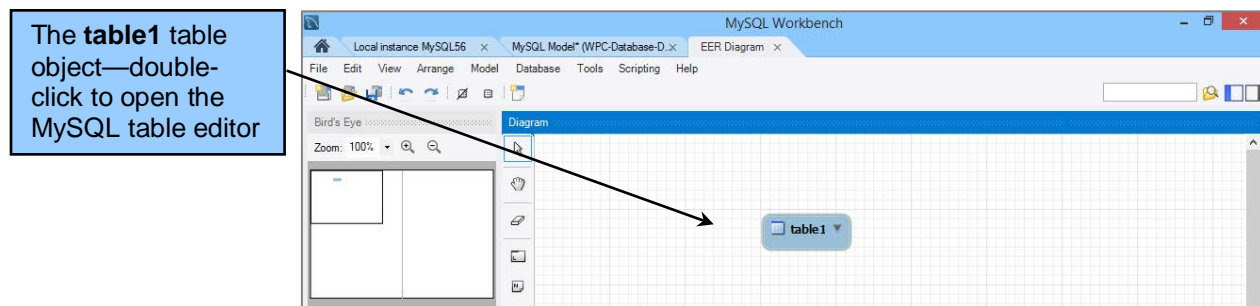


Figure C-21 — The table1 Table Object

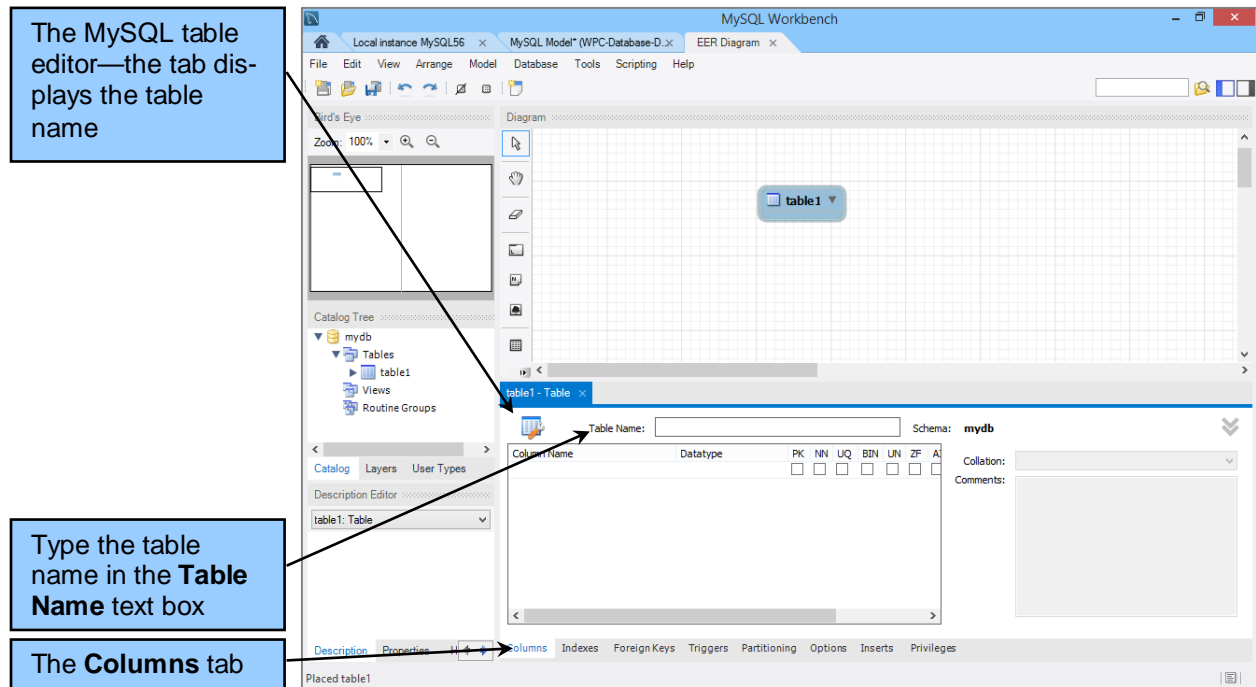


Figure C-22 — The MySQL Table Editor

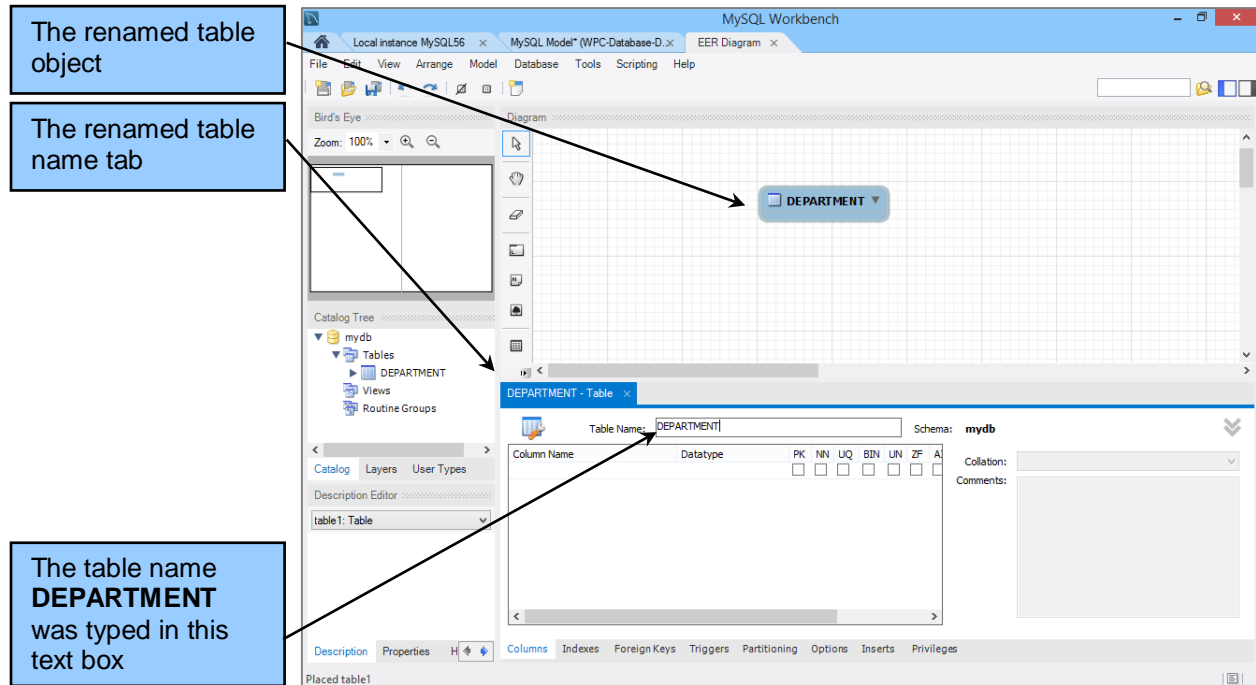


Figure C-23 — The Renamed Table

Now that we have the DEPARTMENT table created and named, we will create the columns in the table.

As shown in Figure C-24, MySQL uses checkboxes with the following abbreviations and a Default text box for defining column characteristics in the MySQL Table Editor:

- **PK** Primary Key – Check if this column is the primary key or part of a composite primary key.
- **NN** NOT NULL – Check if this column must have an inserted value.
- **UQ** UNIQUE – Check if this column must contain a unique value.
- **BIN** Binary – Check if this the column uses only two values, such as 0 and 1 or Yes and No.
- **UN** Unsigned Data Type – Check if this column uses numbers without negative values and you specifically want to not permit negative numbers where they might not otherwise be allowed. Zero fill (ZF) numbers are automatically checked UN.
- **ZF** Zero Fill – Check if this column should be automatically filled with zeros.
- **AI** AUTO\_INCREMENT – Check if this column is a primary key that should have sequential surrogate key values.

The default text is used, of course, for specifying the DEFAULT value for a column if one is required. Figure C-27 shows the complete DEPARTMENT table. The key symbol indicates the primary key, and the filled-in, light blue diamonds indicate NOT NULL columns (NULL columns have an empty diamond).

#### *Creating the DEPARTMENT Table Columns in the MySQL E-R (EER) Diagram:*

1. In the MySQL Table Editor, double-click the row area immediately in the **Column Name** column in the Columns area. MySQL Workbench generates a primary key name idDEPARTMENT. This primary key is also displayed in the DEPARTMENT table object, as shown in Figure C-24.
2. The correct primary key column name for DEPARTMENT (as shown in the CREATE TABLE DEPARTMENT SQL statement shown previously in this appendix) is DepartmentName. Edit the Column Name to read **DepartmentName**.
3. The Datatype for DepartmentName is currently INT, but it should be Char(35). Click the Datatype drop-down list arrow in the Datatype field to display the Datatype drop-down list as shown in Figure C-25. Select the Char datatype from the list (it is at the bottom of the list, and not visible in Figure C-25) and edit the datatype to read **Char(35)**.
  - **NOTE:** Alternatively, you can just type the correct data type into the Datatype field.
4. There are no other settings that need to be changed for the DepartmentName column. It is already marked as PK (Primary Key) and NN (NOT NULL).
5. Double-click the Column Name field in the blank row immediately below DepartmentName. The completed DepartmentName column row, and a new blank column row is displayed as shown in Figure C-26. Note that the correct column settings are now displayed in the DEPARTMENT table object in the E-R diagram.
6. Add the rest of the columns to the DEPARTMENT table. The results are shown in Figure C-27.
7. Click the **Save Model to Current File** button to save the changes, and then close the MySQL Table Editor.

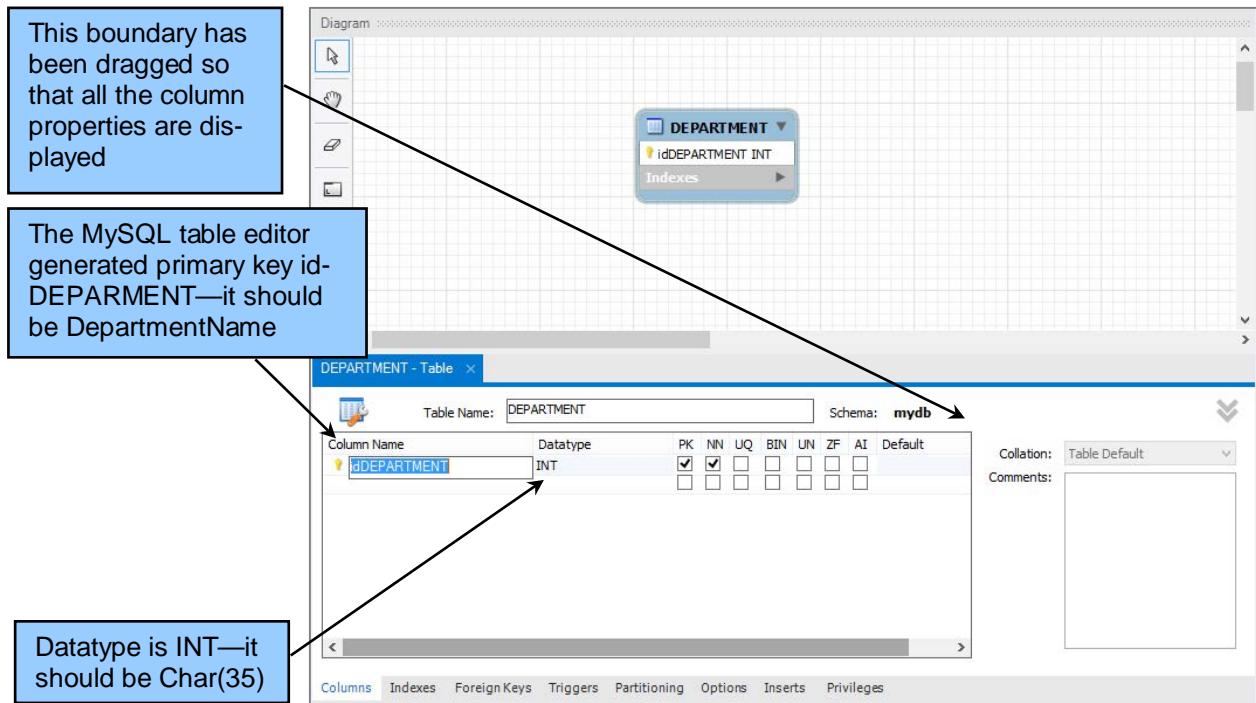


Figure C-24 — The MySQL Table Editor Generated Primary Key idDEPARTMENT

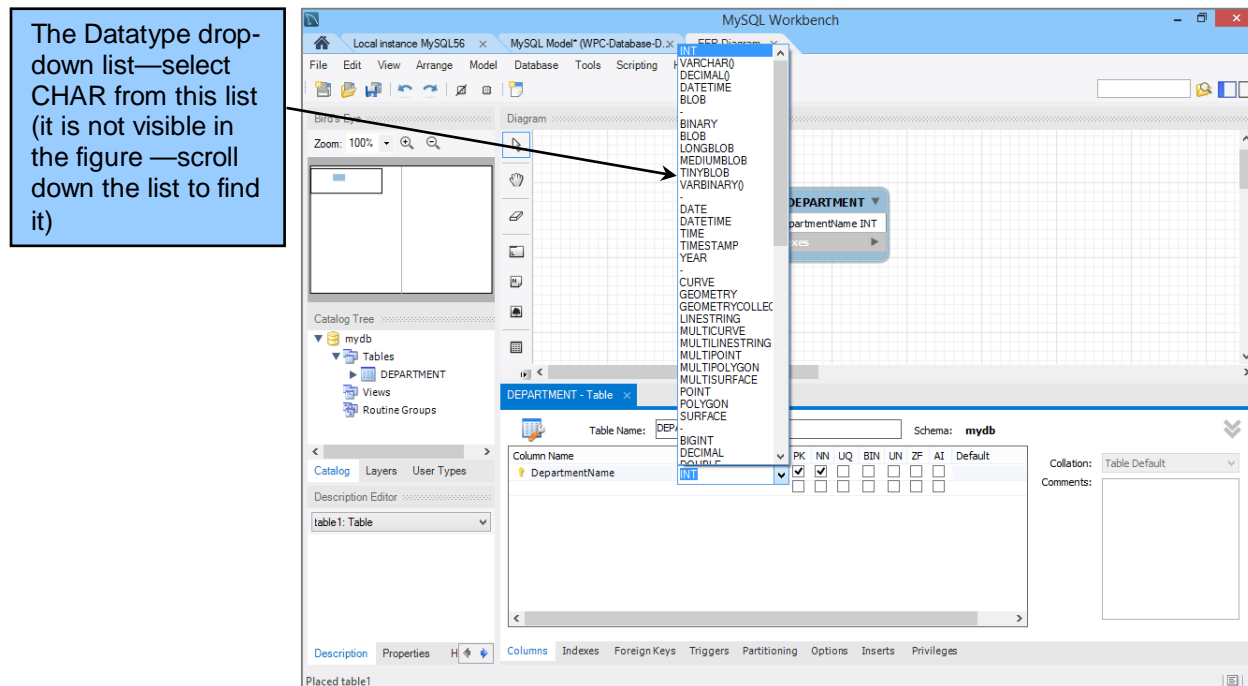


Figure C-25 — The Datatype Drop-Down List

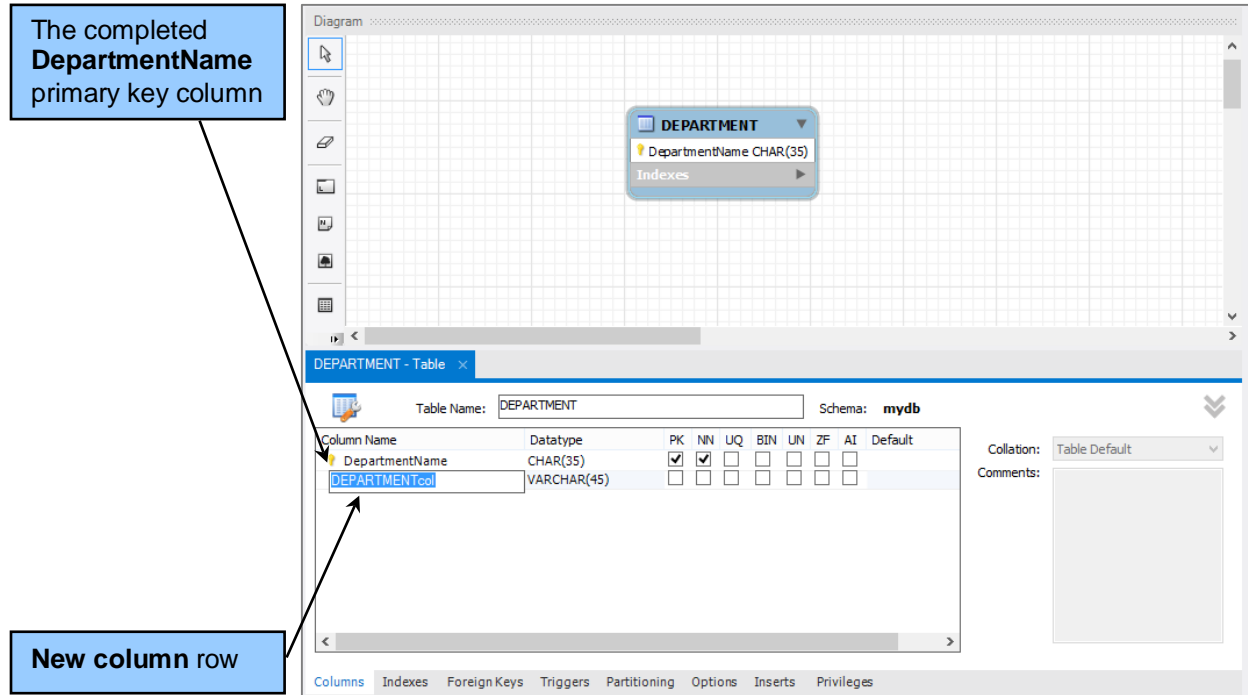


Figure C-26 — The Completed DepartmentName Primary Key Column

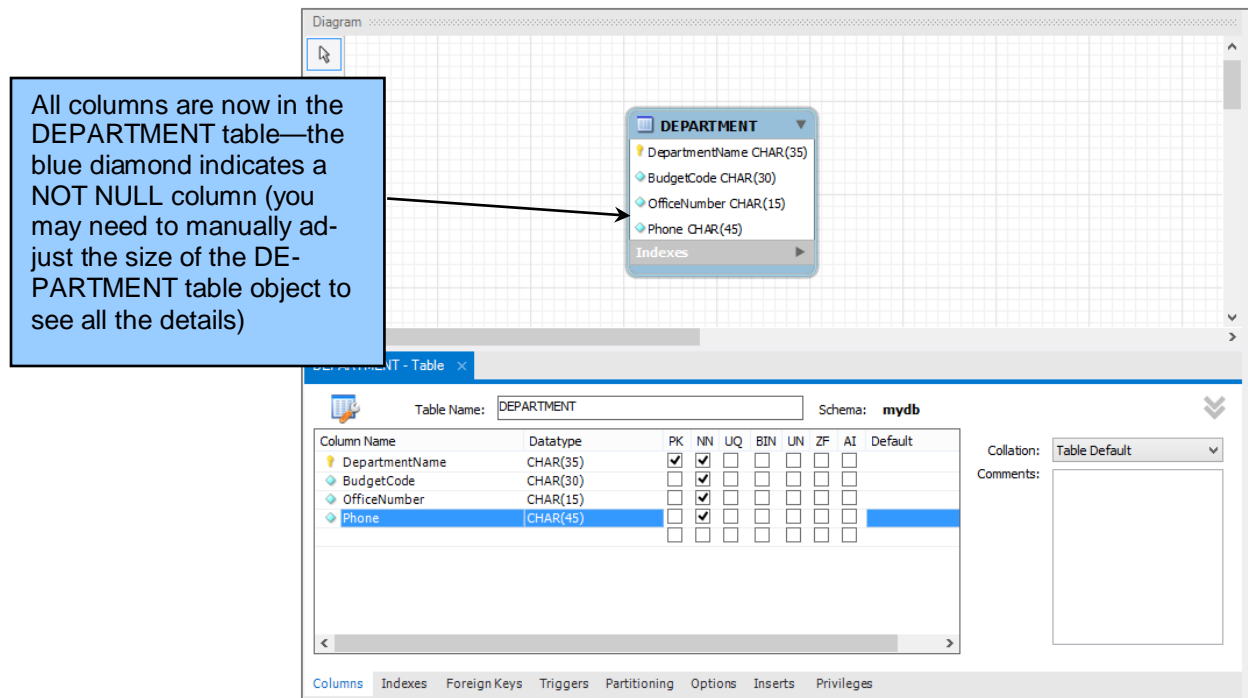


Figure C-27 — The Completed DEPARTMENT Table

Now we will build the EMPLOYEE and PROJECT tables, but we will wait to build the ASSIGNMENT table until we discuss how to create relationships. The process is similar to the process we used to build the DEPARTMENT table, and the results are shown in Figure C-28. Note that the table objects have been resized and rearranged. After you have completed the three tables, close the MySQL Table Editor so your screen looks like Figure C-28.

Now that we have created the tables, we need to connect them with relationships. As shown in Figure C-29, the MySQL Workbench has buttons (labeled in IE Crow's Foot notation) to create a variety of relationships:

- **1:1 Non-identifying Relationship**—Used between two strong entities.
- **1:N Non-identifying Relationship**—Used between two strong entities.
- **1:1 Identifying Relationship**—Used between a strong entity and an ID-dependent weak entity—see discussion below.
- **1:N Identifying Relationship**—Used between a strong entity and an ID-dependent weak entity.
- **N:M Identifying Relationship**—Used between two strong entities—see discussion below.
- **Place a Relationship Using Existing Columns**—See discussion below.

The usage of **1:1 non-identifying**, **1:N non-identifying**, and **1:N identifying** relationships are standard and correct. However, by definition an identifying relationship has to be used in a 1:N relationship (see Chapter 4), so the **1:1 identifying** relationship does not make sense. Similarly, pure N:M relationships

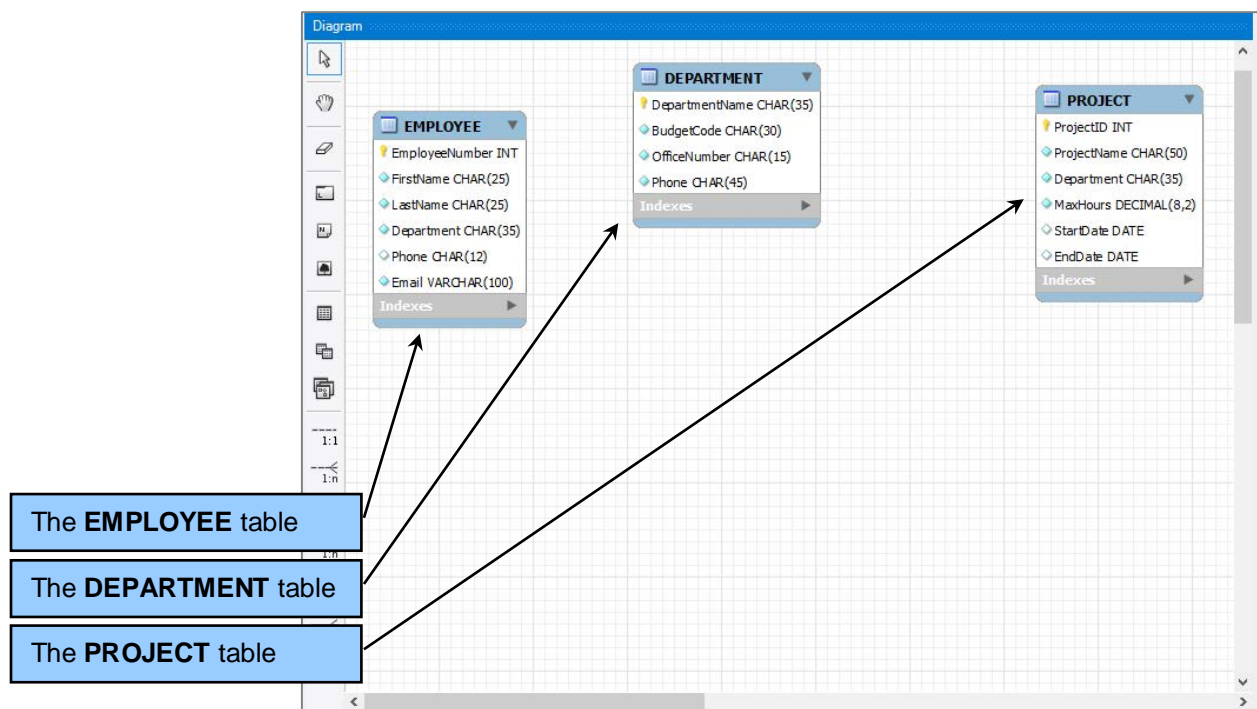


Figure C-28 — The Completed DEPARTMENT, EMPLOYEE and PROJECT Tables

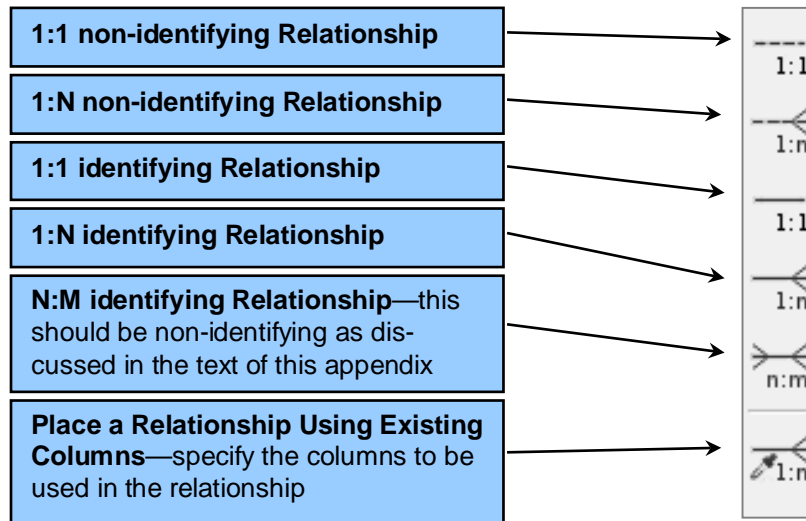


Figure C-29 — Relationships in the MySQL Workbench EER Diagram Toolbar

only exist in data models, and they are always *non-identifying* relationships between two strong entities, so this symbol uses a dashed line instead of a solid one. However, the MySQL Workbench lets us edit relationships after we've created them so we can change any parameter we want to change.

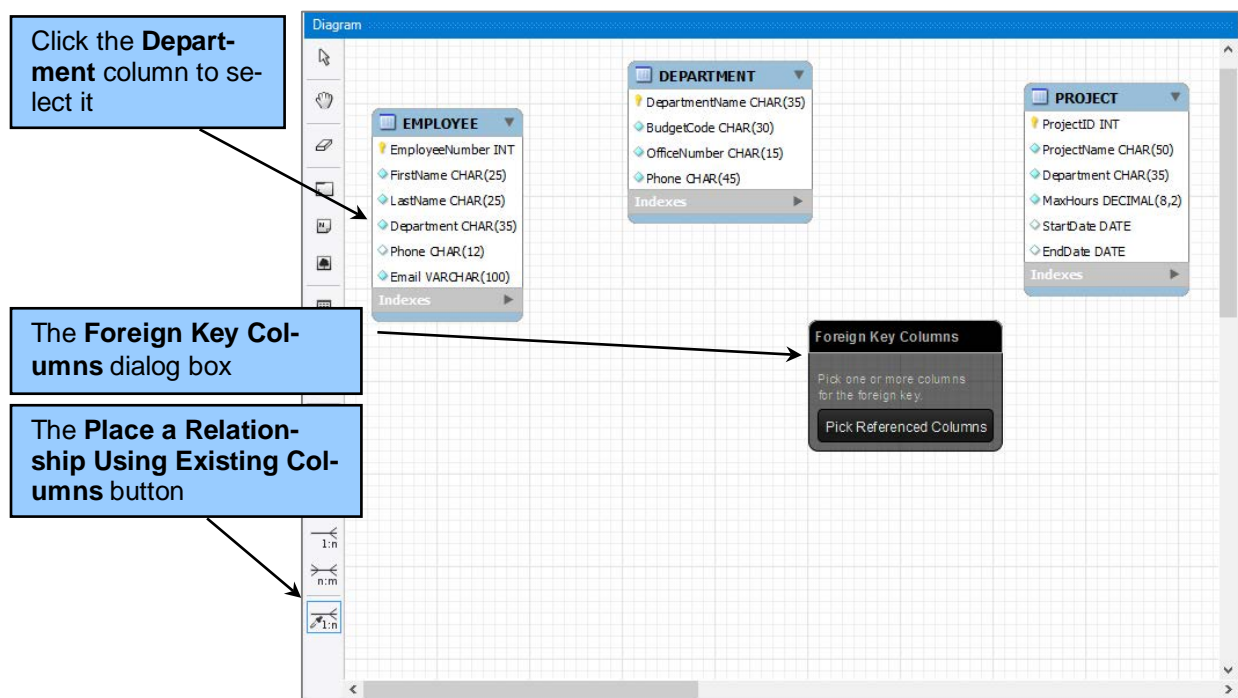
### By The Way

The MySQL Workbench uses the term **non-identifying** relationship, whereas in *Database Concepts* we use the term **nonidentifying** relationship. We have seen the term **non identifying** relationship used in other contexts. All three terms mean exactly the same thing, and which is used is a matter of style. Since MySQL Workbench uses *non-identifying*, we will also use that term in this appendix for consistency with the MySQL Workbench screen shots, while remaining well aware that we have used *nonidentifying* in *Database Concepts* itself.

Among the buttons that MySQL Workbench provides for creating relationships in the database design, the **Place a Relationship Using Existing Columns** choice is very useful. Normally, when we create a relationship, the MySQL Workbench automatically adds a foreign key, even if the column that should be the foreign key is already there! And this relationship usually turns out to be a non-identifying relation, despite the solid line shown on the button. Since both EMPLOYEE and PROJECT contain the column that will be the foreign key—in both cases it is the Department column—we will use this button to create the relationships between the tables currently in the E-R diagram. We will start with the relationship between DEPARTMENT and EMPLOYEE.

**Creating a 1:N Nonidentifying Relationship Between Two Tables:**

1. Click the **Place a Relationship Using Existing Columns** button.
2. MySQL Workbench displays a Foreign Key Columns dialog box instructing us to *Pick one or more columns for the foreign key*, as shown in Figure C-30.
3. Click the **Department** column in EMPLOYEE to select it.
4. Click the **Pick Referenced Columns** button in the Foreign Key Columns dialog box, as shown in Figure C-31.
5. The Foreign Key Columns dialog box becomes the Referenced Columns dialog box, instructing us to *Pick matching columns for the referenced table*, as shown in Figure C-32.
6. Click the **DepartmentName** column in DEPARTMENT. A 1:N non-identifying relationship is created between DEPARTMENT (parent-the 1 side of the relationship) and EMPLOYEE (child-the N side of the relationship), as shown in Figure C-33. Note the red diamond indicating a foreign key in EMPLOYEE.
7. To see the details of the relationship, right-click the relationship line, then click Edit Relationship, and then click the **Foreign Key tab**. The relationship parameters are displayed as shown in Figure C-34.

**Figure C-30 — The Foreign Key Columns Dialog Box – Pick Columns for Foreign Key**



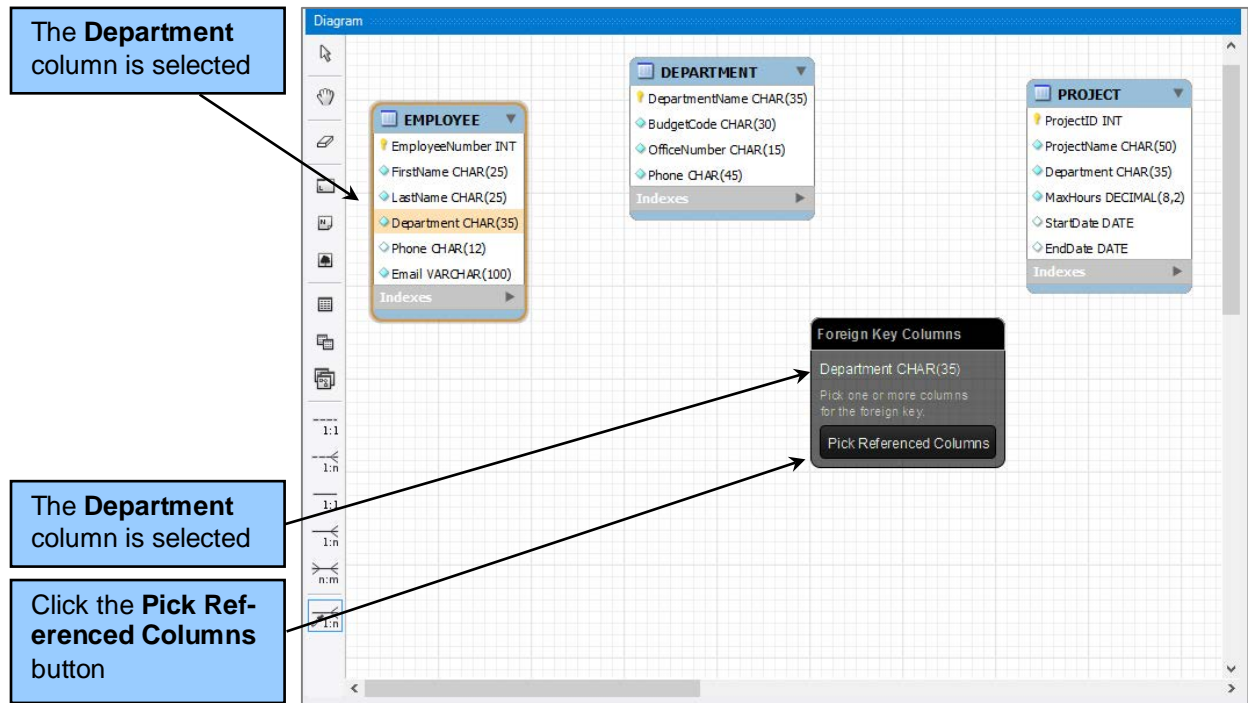


Figure C-31 — EMPLOYEE.Department Selected As The Foreign Key Column

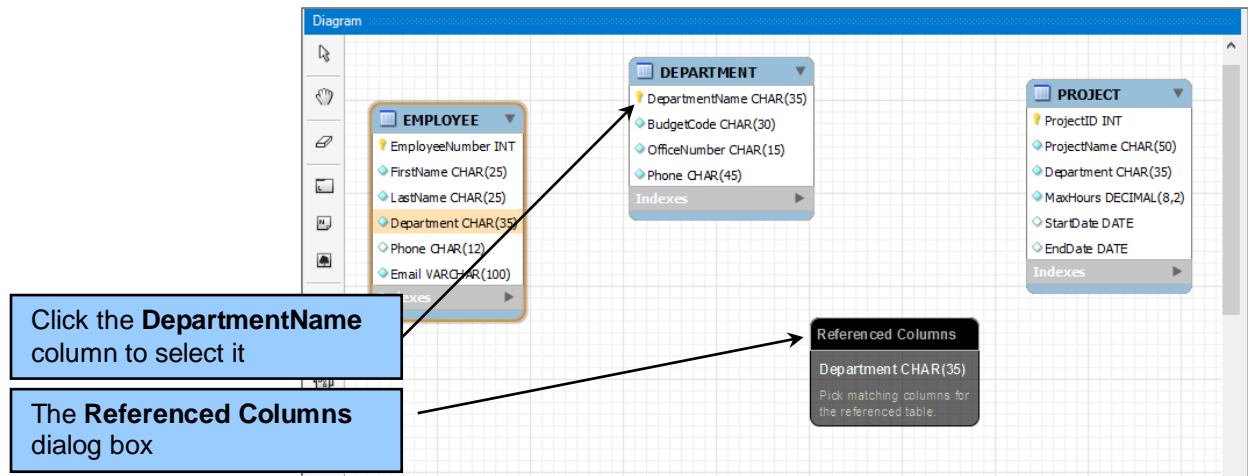
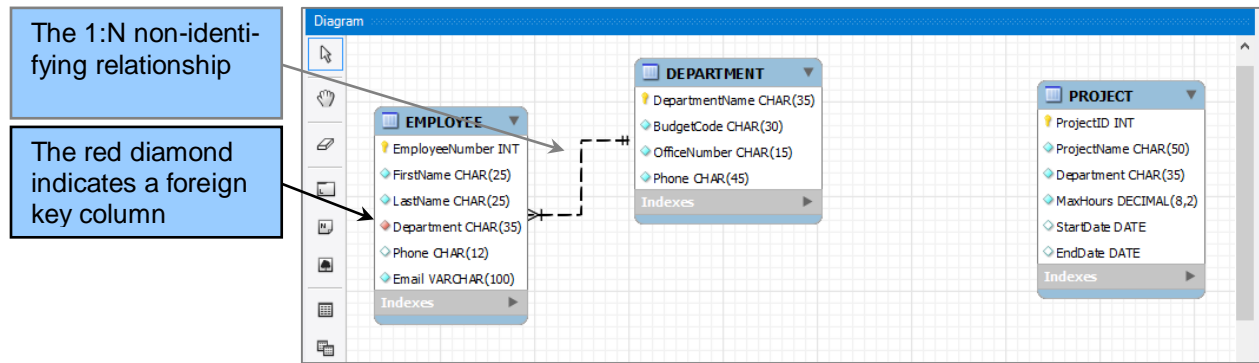


Figure C-32— The Referenced Columns Dialog Box – Pick Referenced Primary Key

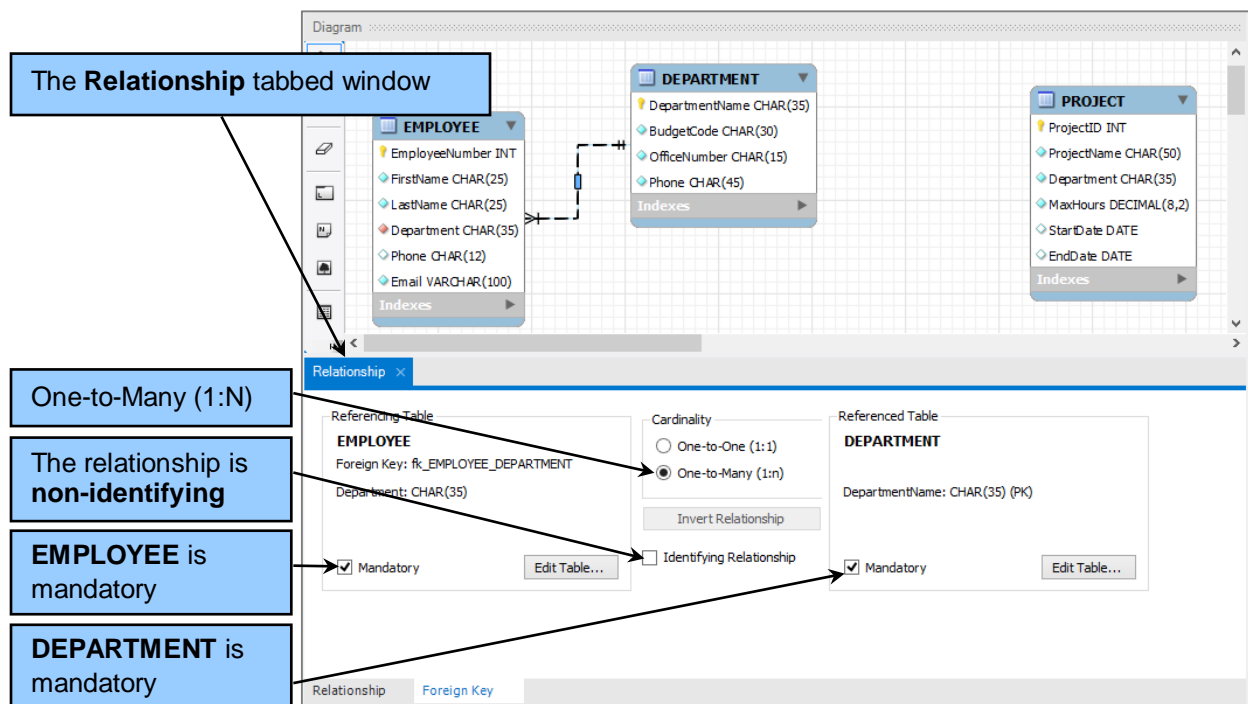
The MySQL Relationship Editor shown in Figure C-34 shows that this relationship (it has been named `fk_EMPLOYEE_DEPARTMENT` by the MySQL Workbench, but we can change this if we want), and can control the name in the SQL code when we actually construct the tables and relationships in our DBMS (in the SQL `CREATE TABLE EMPLOYEE` statement in Figure C-13 we used a `FOREIGN KEY CONSTRAINT` phrase to name this corresponding foreign key constraint as `EMP_DEPART_FK`).



**Figure C-33 — The Completed 1:N Non-identifying Relationship**

The options on the Foreign Key page shown in Figure C-34 allow us to control all aspects of the relationship. As shown there, the relationship is one-to-many (1:N), non-identifying, with both DEPARTMENT and EMPLOYEE having mandatory participation in the relationship (i.e., minimum cardinality of 1). The non-identifying 1:N parameters are correct, but what about the minimum cardinalities?

Does a DEPARTMENT have to have at least one employee? This is actually a business rule question, but we will assume that the answer for WPC is yes, and that WPC does not allow departments without employees to exist.



**Figure C-34 — The Foreign Key Properties**

Does an EMPLOYEE have to be assigned to a department? Again, this is a business rule question, but the fact that EMPLOYEE.Department is NOT NULL with a DEFAULT value of Human Resources is a good indication that the answer for WPC is yes, and that WPC does not allow employees unassigned to departments to exist.

Therefore the mandatory parameters are correct, and our relationship is correct as drawn.

8. Close the MySQL Relationship Editor by click the **Close [X]** button on the tab.

We can create the relationship between DEPARTMENT and PROJECT the same way. We will assume that every PROJECT must have a sponsoring DEPARTMENT, but that a DEPARTMENT is not required to have any PROJECTs in the works. The E-R diagram with the additional relationship is shown in Figure C-35.

At this point we still need to build the ASSIGNMENT table and its relationships with EMPLOYEE and PROJECT. However, because ASSIGNMENT is an association table in an associations relationship with EMPLOYEE and PROJECT (association relationships are discussed in Chapter 5), and since association tables are just intersection tables with additional attributes (again, as discussed in Chapter 5), we will use this as an opportunity to illustrate how MySQL Workbench handles N:M relationships.

As discussed in Chapters 4 and 5, an N:M relationship only exists in a data model (as a non-identifying relationship between two strong entities). In a database design, the N:M relationship becomes two 1:N ID-dependent identifying relationships linking the two original tables through a new, third table called

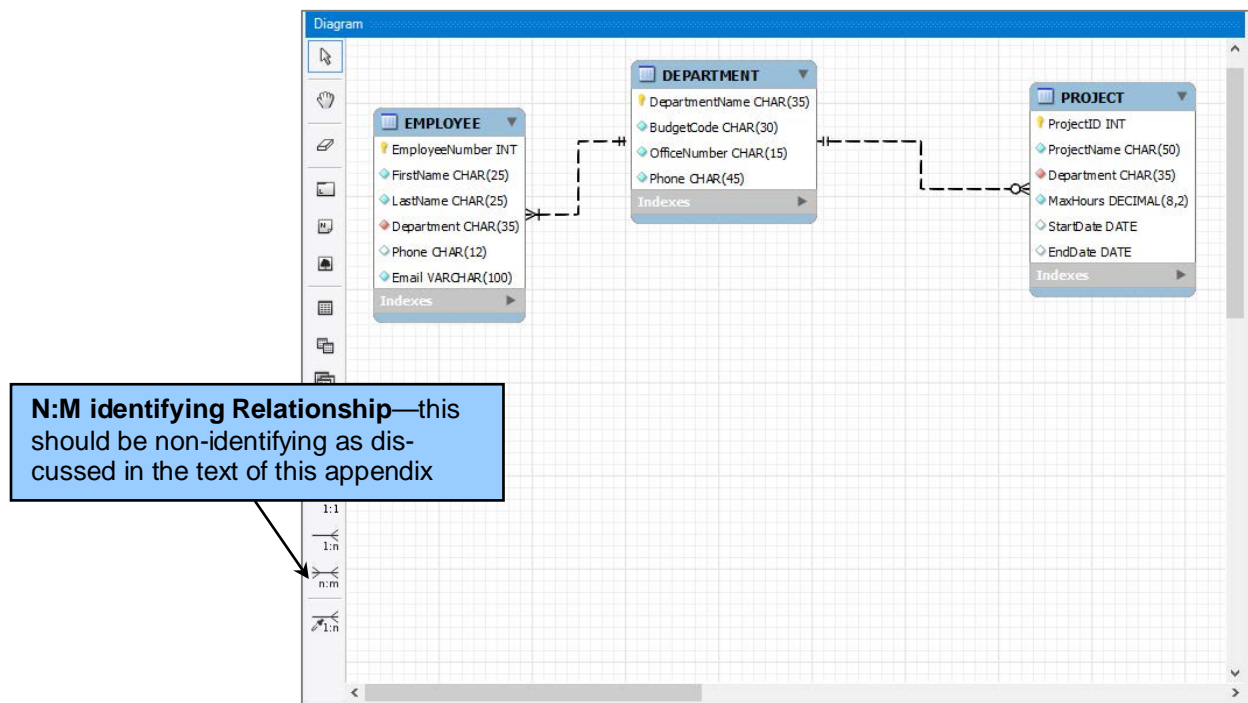


Figure C-35 — The Completed DEPARTMENT, EMPLOYEE and PROJECT Relationships

an intersection table. However, MySQL Workbench only builds database designs, and will automatically create the intersection table with the two 1:N relationships whenever we specify an N:M relationship.

### Creating an N:M Nonidentifying Relationship Between Two Tables:

1. Click the **N:M identifying Relationship** button, then click the **EMPLOYEE** table, and then click the **PROJECT** table.
2. As shown in Figure C-36, the MySQL Workbench creates an *intersection table* named **EMPLOYEE\_has\_PROJECT**, and places it in the E-R diagram together with two 1:N ID-dependent identifying relationships between (1) **EMPLOYEE\_has\_PROJECT** and **EMPLOYEE**, and (2) **EMPLOYEE\_has\_PROJECT** and **PROJECT**.

We have just demonstrated that the MySQL Workbench can only create database designs, not data models, by showing how MySQL Workbench automatically converts a M:N relationship into two 1:N relationships to a intersection table . Now we simply have to do some editing in the MySQL Table Editor:

- Change the table name to **ASSIGNMENT**.
- Change the primary key attribute names to **EmployeeNumber** and **ProjectID**.
- Add the **HoursWorked** attribute.

What about the relationship cardinalities? We will assume that every project has to have at least one employee assigned to it, and that every employee has to work on at least one project. Therefore the minimum cardinalities are correct as created. The final E-R diagram is shown in Figures C-36 and C-37, and, except for the exact arrangement of the tables and relationships, is the same as the WPC database design shown in Figures 5-16 and 5-17. And that completes our introduction to MySQL.

3. Save the WPC-Database-Design in the MySQL Workbench, and close the EER Diagram window.
4. Close the MySQL Model window.
5. Close the MySQL Workbench.

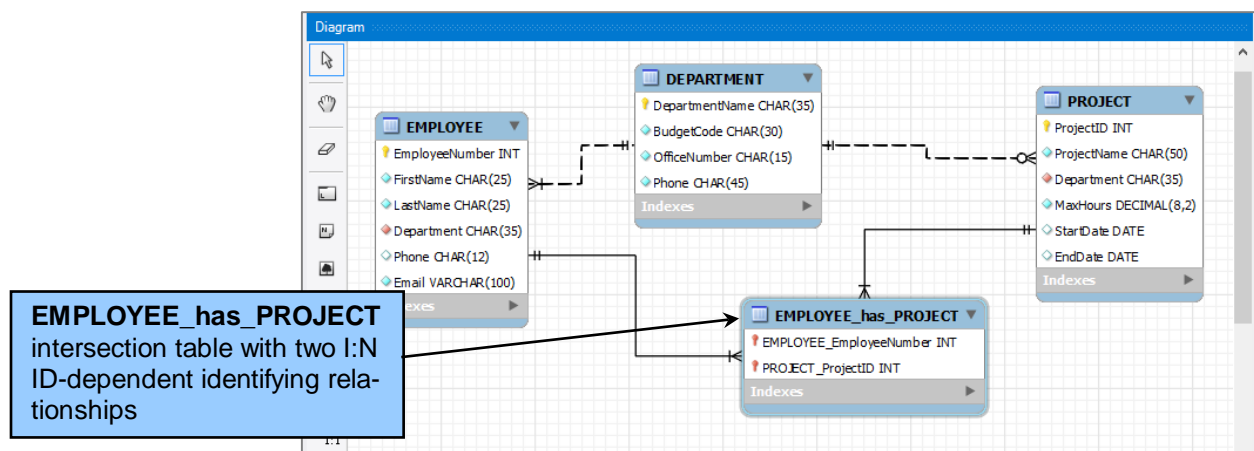


Figure C-36 — The Completed Set of Relationships

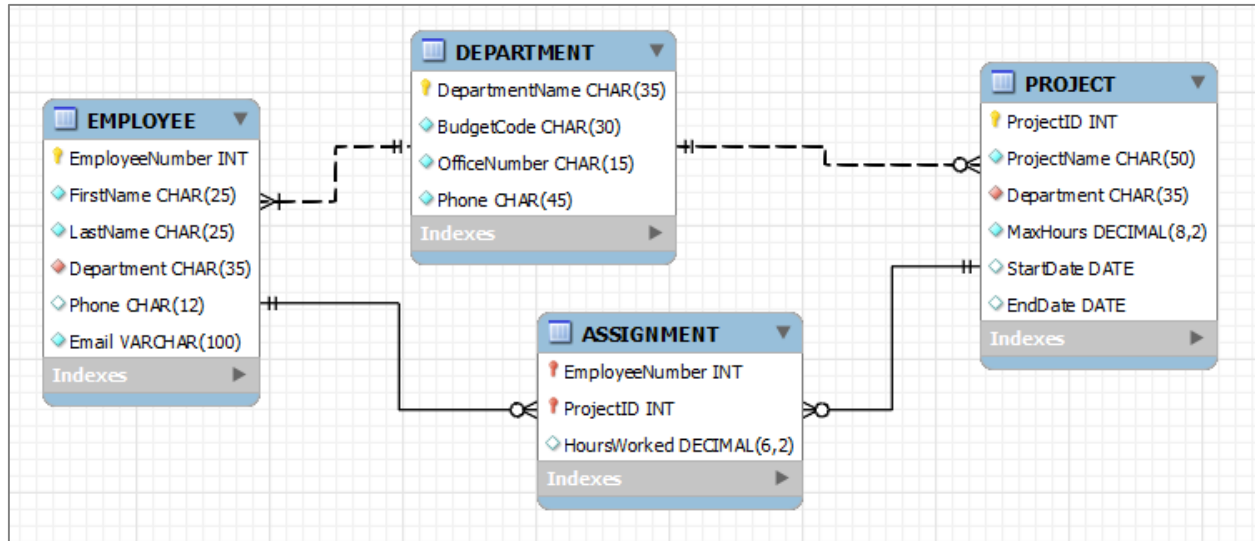


Figure C-37 — The Completed E-R Diagram

### Key Terms

*.sql	Connect to Database dialog box
Connect to MySQL Server dialog box	Data model
Database design	Default schema
Execute SQL Script in Connected Server button	Execute Current SQL Statement in Connected Server button
MySQL Connector/ODBC	MySQL Community Server
MySQL Workbench	Object browser
Open Connection to Start Querying	Place a New Table button
Place a Relationship Using Existing Columns button	Result(1) window
Root	Schema
SQL script	

### Review Questions

- C.1 What is MySQL?
- C.2 What is the primary advantage of using MySQL instead of SQL Server Express Edition?

- C.3 What are the three MySQL programs that are recommended as a useful set of MySQL software products?
- C.4 What are two purposes of the MySQL Workbench?
- C.5 What is a MySQL schema?
- C.6 How do you create a new database in MySQL?
- C.7 What is a MySQL default schema? How do you specify the default schema?
- C.8 What is an SQL script? What types of SQL statements and commands can be run more efficiently as scripts?
- C.9 What tool(s) can you use to create an SQL script?
- C.10 What file extension should you use with MySQL scripts?
- C.11 How do you create, save, and run an SQL script in MySQL?
- C.12 How does MySQL use the `AUTO_INCREMENT` keyword? What limitations are there on the use of the `AUTO_INCREMENT` keyword?
- C.13 How do you create and run an SQL query in MySQL?
- C.14 How do you obtain online help when using the MySQL Workbench? Where is the help information displayed?
- C.15 What is the purpose of the MySQL Connector/OBDC? How do you install it, and where does it appear after it is installed?
- C.16 The MySQL Workbench creates *database models*. What is a data model? What is a database design? Which does MySQL Workbench create?
- C.17 How do you create a database model in MySQL Workbench?
- C.18 How do you create a new E-R diagram in the MySQL Workbench?
- C.19 How do you create a table in an E-R diagram?
- C.20 How do you create a 1:N non-identifying relationship in an E-R diagram?
- C.21 How does the MySQL Workbench handle N:M relationships?

## Exercises

---

- C.22 If you haven't already done so, download and install MySQL 5.6 Community Server and the MySQL Workbench. Use the default setting for each installation. When you are asked for a password for the Root user account, create one.
- C.23 If you haven't already done so, work through the steps described in this appendix to create and populate the WPC database.
- C.24 Use MySQL and the MySQL Workbench to create and run the following SQL queries in the "SQL for Relational Queries" section of Chapter 3:
- SQL-Query-CH03-01 through SQL-Query-CH03-31
  - SQL-Query-CH03-34 through SQL-Query-CH03-49
  - SQL-Query-CH03-55 through SQL-Query-CH03-56
- Save each query as follows:
- Create and run each query in the MySQL Workbench.
  - After you have run each query, use the **File | Save Script As** command to save the query. By default, MySQL saves each file as an \*.sql script. Name your queries in numerical sequence, starting with the file name MySQL-SQL-Query-CH03-01.sql.
- C.25 Use MySQL and the MySQL Workbench to run one or more of the saved SQL queries you created in question C.24:
- At this point, you should have only one SQL Script window open. Open a query with the **File | Open SQL Script** command.
  - Experiment with opening and closing SQL Script windows and running various queries in these windows.
- C.26 Complete exercise 3.63 using MySQL. Start each saved query name with MySQL- and use the default \*.sql file extension. (The first saved query name should be MySQL-SQL-Query-AWE-3-1-A.sql.)
- C.27 Complete exercise 3.64 using MySQL. Start the saved query name with MySQL- and use the default \*.sql file extension. The saved query name should be MySQL-SQL-Query-AWE-3-3-E.sql.
- C.28 The database design for the Wallingford Motor Customer Relationship Management database is shown in Figure AW-5-1. In the MySQL Workbench, create a new MySQL Database Model named WMCRM-Database-Design. In this model, add an E-R diagram, and recreate the database design shown in Figure AW-5-1 in this diagram.

- C.29 The database design for Heather Sweeney Designs is shown in Figure 5-27. In the MySQL Workbench, create a new MySQL Database Model named HSD-Database-Design. In this model, add an E-R diagram, and recreate the database design shown in Figure 5-27 in this diagram.

