# **Database Processing**

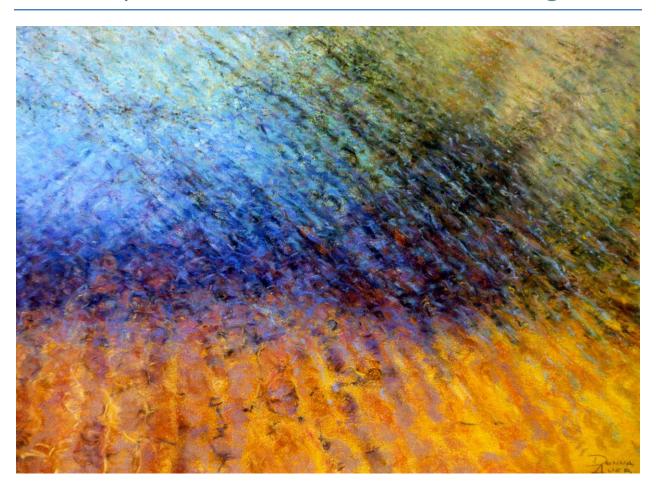
Fundamentals, Design, and Implementation

14th Edition

David M. Kroenke • David J. Auer

# Online Appendix E

Getting Started with MySQL 5.6 Workbench Data Modeling



Vice President, Business Publishing: Donna Battista

**Editor in Chief:** Stephanie Wall **Acquisitions Editor:** Nicole Sam

Program Manager Team Lead: Ashley Santora

**Program Manager:** Denise Weiss **Editorial Assistant:** Olivia Vignone

Vice President, Product Marketing: Maggie Moylan Director of Marketing, Digital Services and Products:

Jeanette Koskinas

Executive Product Marketing Manager: Anne Fahlgren

Field Marketing Manager: Lenny Ann Raper Senior Strategic Marketing Manager: Erin Gardner Product Marketing Assistant: Jessica Quazza Project Manager Team Lead: Jeff Holcomb

Project Manager: Ilene Kahn Operations Specialist: Diane Peirano Senior Art Director: Janet Slowik Text Designer: Integra Software Services Pvt. Ltd. Cover Designer: Integra Software Services Pvt. Ltd. Cover Art: Donna Auer

Vice President, Director of Digital Strategy &

Assessment: Paul Gentile

Manager of Learning Applications: Paul Deluca

Digital Editor: Brian Surette

Digital Studio Manager: Diane Lombardo
Digital Studio Project Manager: Robin Lazrus
Digital Studio Project Manager: Alana Coles
Digital Studio Project Manager: Monique Lawrence
Digital Studio Project Manager: Regina DaSilva

**Full-Service Project Management** 

and Composition: Integra Software Services Pvt. Ltd.

Printer/Binder: RRD Willard

**Cover Printer:** Phoenix Color/Hagerstown **Text Font:** 10/12 Mentor Std Light

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 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.

Oracle Database 12c and Oracle Database Express Edition 11g Release  $2\ 2014$  by Oracle Corporation. Reprinted with permission. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Mozilla 35.104 and Mozilla are registered trademarks of the Mozilla Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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.xt. This book is not sponsored or endorsed by or affiliated with The PHP Group.

Copyright © 2016, 2014, 2012 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.

Database processing: fundamentals, design, and implementation/David M. Kroenke, David J. Auer.—Fourteenth edition. pages cm

Includes bibliographical references and index.

ISBN 978-0-13-387670-3 (student edition)-ISBN 978-0-13-387676-5

(instructor's review copy)

1. Database management. I. Auer, David J. II. Title.

QA76.9.D3K76 2016

005.74-dc23

2015005632

10 9 8 7 6 5 4 3 2 1



ISBN 10: 0-13-387670-5 ISBN 13: 978-0-13-387670-3

Appendix E — 10 9 8 7 6 5 4 3 2 1

#### **Chapter Objectives**

- Learn how to install MySQL Workbench and MySQL ODBC/Connector
- Learn how to create database designs in the MySQL Workbench

#### What is the Purpose of this Appendix?

One widely used edition of MySQL 5.6 is the **MySQL Community Server**, which 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, only to have Oracle Corporation acquire Sun Microsystems (the deal was finalized on January 27<sup>th</sup>, 2010, after European Commission approval on January 21<sup>st</sup>, 2010—see <a href="http://www.oracle.com/us/sun/index.htm">http://www.oracle.com/us/sun/index.htm</a>). The Oracle corporation now owns MySQL in addition to its flagship Oracle Database product. 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

<sup>&</sup>lt;sup>1</sup> For information about Oracle Database 11g Release 2, see on-line Chapter 10B.

server. The MySQL Workbench graphical user interface (GUI) utility is the main user interface when using MySQL. The MySQL Connector/ODBC is the ODBC driver for the MySQL DBMS. Open Database Connectivity (ODBC) and OBDC Drivers are discussed in Chapter 11. You will need the MySQL Connector/ODBC if you are using the MySQL DBMS, and you want to create the Web database applications and Web pages discussed in Chapter 11.

The purpose of this Appendix is to:

- Learn how to use the MySQL Workbench to create database designs as discussed in Chapter 6.
- Learn how to install the MySQL Connector/ODBC for use with Web database applications as discussed in Chapter 11.

Installation and administration of MySQL 5.6, including developing and using databases, is discussed in Chapter 10C. MySQL and its associated utilities are very easy to install. As described in Chapter 10C, install and configure MySQL Server Community Edition 5.6, the MySQL workbench and the MySQL Connector/ODBC. If you are running the Windows operating system, you should do this, as described in Chapter 10C, by using the MySQL Installer for Windows.

#### Why Should I Learn to Use the MySQL Workbench for Database Design?

The MySQL Workbench contains a good set of features for creating database designs. If you are working with the MySQL DBMS, this is a natural pairing, and you will be using the same GUI for database design and implementation. Even if you aren't using the MySQL DBMS, the database design capabilities of the MySQL Workbench make it well worth considering as your database design tool.

#### What Will This Appendix Teach Me?

This appendix is designed to get you started creating database designs using the MySQL Workbench, and discuss the installation of the MySQL Connector/ODBC.

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

The MySQL Workbench has many capabilities beyond those described in Chapter 10C and this Appendix. It is possible, for example, to use the MySQL Workbench to reverse engineer (discussed in Chapter 8) MySQL databases. It is also possible to create a database in MySQL from a database design in MySQL Workbench. For more information on the capabilities of the MySQL Workbench, see the MySQL Workbench documentation available at <a href="http://dev.mysql.com/doc/workbench/en/">http://dev.mysql.com/doc/workbench/en/</a>.

### How Do I Install the MySQL Workbench and the MySQL Connector/OBDC?

Unlike the installation of Microsoft Access 2013 (separately or as part of Office 2013), or Microsoft SQL Server 2014, or Oracle Database, where the GUI management utility and ODBC support is installed as part of the DBMS installation, you *may* have to take extra steps to install the MySQL Workbench and ODBC support

for MySQL if you are not using the Windows operating system. Fortunately, MySQL and its associated utilities are very easy to install. A full discussion of the installation and administration of MySQL 5.6, including developing and using databases, is in on-line Chapter 10C.

If you are using a version of the Microsoft Windows operating system, you should download, install and use the **MySQL Installer for Windows**, which is available at <a href="http://dev.mysql.com/tech-resources/articles/mysql-installer-for-windows.html">http://dev.mysql.com/tech-resources/articles/mysql-installer-for-windows.html</a>. Using the MySQL Installer for Windows, you can select the MySQL Workbench and the MySQL Connector/ODBC as part of your MySQL Installation. The MySQL Installer for Windows is also used to update installed MySQL components.

In you are using any other operating system, you will need to download and install the appropriate versions of the MySQL Workbench and the MySQL Connector/ODBC. The MySQL Workbench can be downloaded from the MySQL Website <a href="http://dev.mysql.com/downloads/tools/workbench/">http://dev.mysql.com/downloads/tools/workbench/</a>, and the MySQL Connector/ODBC from <a href="http://dev.mysql.com/downloads/connector/odbc/">http://dev.mysql.com/downloads/connector/odbc/</a>. After you have downloaded the files, run the appropriate installation routine. As soon as the installation is complete, the MySQL Workbench and the MySQL ODBC 5.3.4 Driver will be available for use.

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

As also discussed in online Chapter 10B, 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 E-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>2</sup> The MySQL Workbench splash screen is displayed, followed by the MySQL Workbench window with the Home page displayed, as shown in Figure E-2.

The MySQL Workbench Home tab is a dashboard allowing us access to MySQL database design (mislabeled 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>3</sup>

E-5

<sup>&</sup>lt;sup>2</sup> Alternatively, we recommend that you pin MySQL Workbench 6.1 CE to the Taskbar, as shown in Figures E-1, 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>&</sup>lt;sup>3</sup> If any models are listed in the Models section, right-click each model name and click the **Remove Model File from List** command.

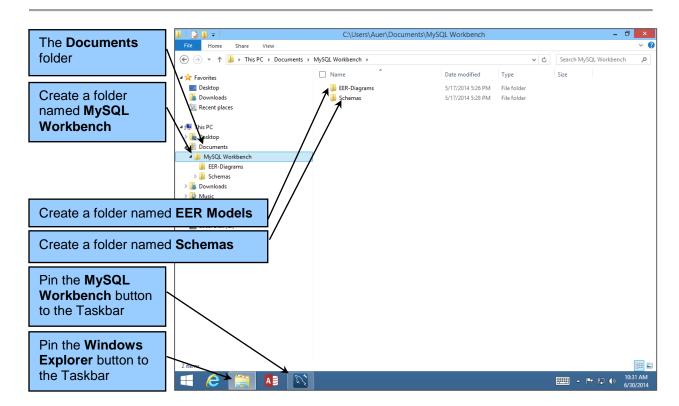


Figure E-1 — The MySQL Workbench Folder in Windows Explorer

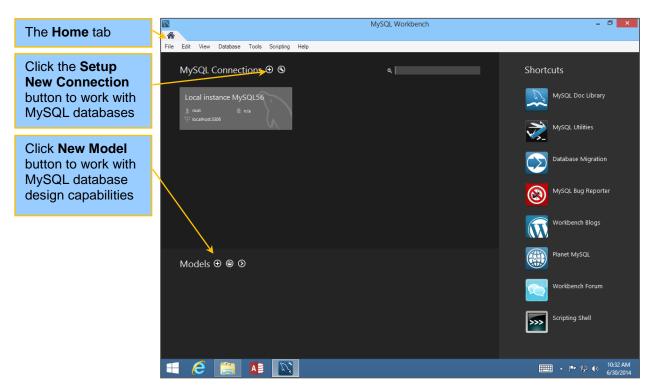


Figure E-2 — The MySQL Workbench Home Tab

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 Create Database Designs in the MySQL Workbench?

As we discuss in Chapters 5 and 6, 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 in Microsoft Access in the Project Question in Chapters 1 and 2, and will create in SQL Server 2014 in Chapter 7.<sup>4</sup> The column characteristics for each of the WPC tables are shown in Figure E-3.

#### 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 5, 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>&</sup>lt;sup>4</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 5 and 6) are taught before using SQL to create the databases (which we cover in Chapter 7). In this case, the database design will precede the actual implementation of the database in the DBMS. We prefer to introduce SQL queries 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 data warehouses or datamarts as discussed in Chapter 12) 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 likes best.

#### DEPARTMENT

Column Name	Туре	Key	Required	Remarks
DepartmentName	Text (35)	Primary Key	Yes	
BudgetCode	Text (30)	No	Yes	
OfficeNumber	Text (15)	No	Yes	
Phone	Text (12)	No	Yes	

#### **EMPLOYEE**

Column Name	Туре	Key	Required	Remarks
EmployeeNumber	AutoNumber	Primary Key	Yes	Surrogate Key
FirstName	Text (25)	No	Yes	
LastName	Text (25)	No	Yes	
Department	Text (35)	No	Yes	
Phone	Text (12)	No	No	
Email	Text (100)	No	Yes	

#### PROJECT

Column Name	Туре	Key	Required	Remarks
ProjectID	Number	Primary Key	Yes	Long Integer
Name	Text (50)	No	Yes	
Department	Text (35)	Foreign Key	Yes	
MaxHours	Number	No	Yes	Double
StartDate	Date/Time	No	No	
EndDate	Date/Time	No	No	

#### ASSIGNMENT

Column Name	Туре	Key	Required	Remarks
ProjectID	Number	Primary Key, Foreign Key	Yes	Long Integer
EmployeeNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
HoursWorked	Number	No	No	Double

Figure E-3 — The WPC Database Table Column Characteristics

#### 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 E-4.
- 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 E-5. 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 this table in Figure E-3, we can see the columns that are used in the DEPARTMENT table.

#### 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 E-5.
- 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 E-6. We do not need to change any of these.
- 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 E-7.
- 4. Double-click the **table1** object to open the MySQL table editor for table1 as shown in Figure E-8. 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 the 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 E-9.
- 6. If necessary, drag the right boundary of the Columns area to the right so that all the column property settings are displayed. Figure E9 shows the results of this operation, where now a check box labeled *AI* and a text box labeled *Default* can be seen.

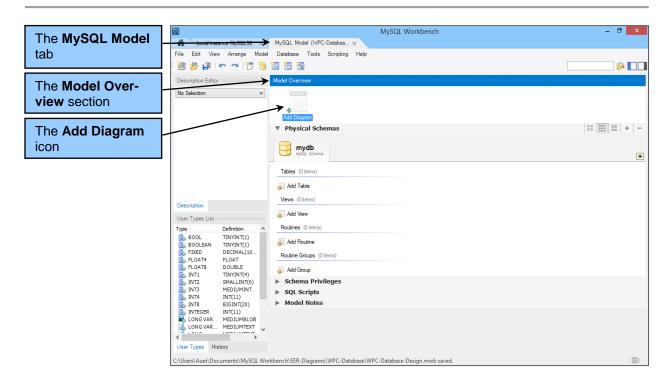


Figure E-4 — The MySQL Model Tab and Window

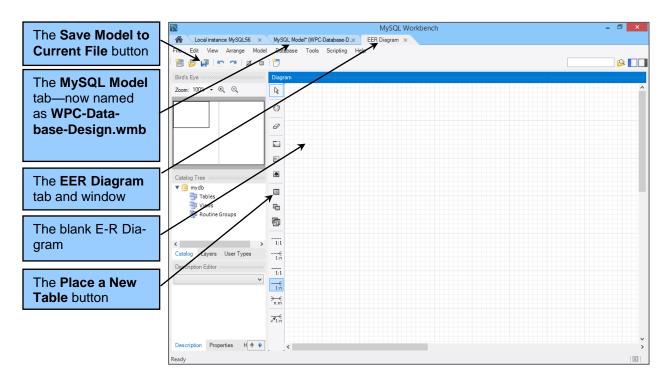


Figure E-5 — The Blank E-R Diagram

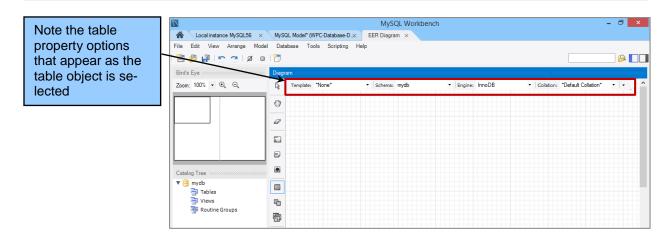


Figure E-6 — Placing the Table Object

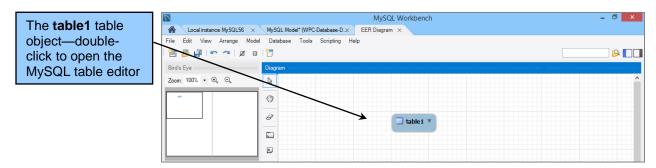


Figure E-7 — The table1 Table Object

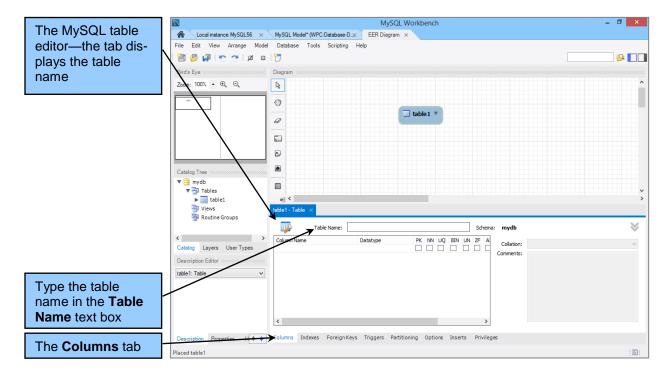


Figure E-8 — The MySQL Table Editor

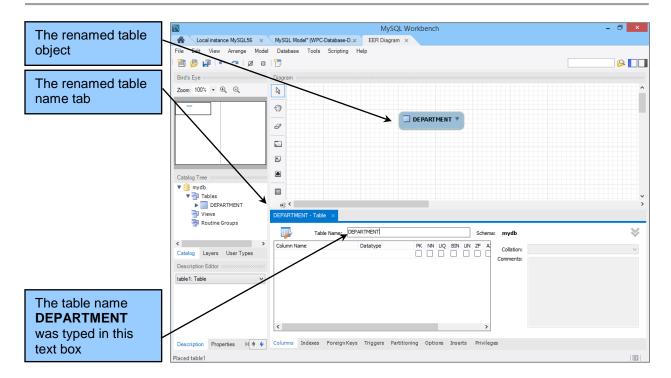


Figure E-9 — The Renamed Table

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

As shown in Figure E-10, 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 E-10 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 E-10.
- 2. The correct primary key column name for DEPARTMENT (as shown in the CREATE TABLE DE-PARTMENT 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 E-11. Select the Char datatype from the list (it is at the bottom of the list, and not visible in Figure E-11) 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 E-12. Note that the correct column settings are now displayed in the DEPARMENT table object in the E-R diagram.

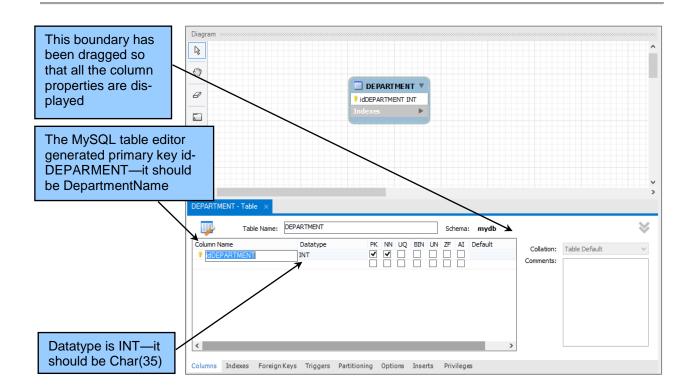


Figure E-10 — The MySQL Table Editor Generated Primary Key idDEPARTMENT

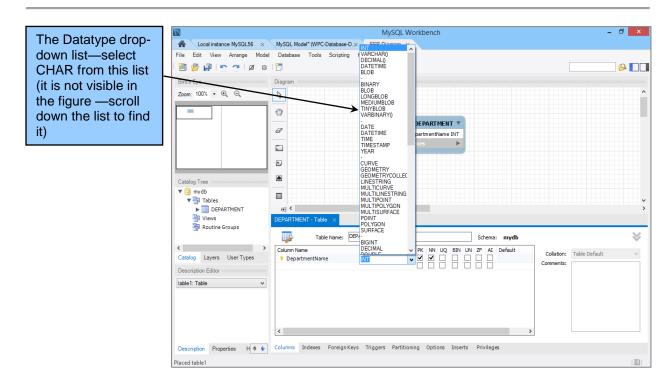


Figure E-11 — The Datatype Drop-Down List

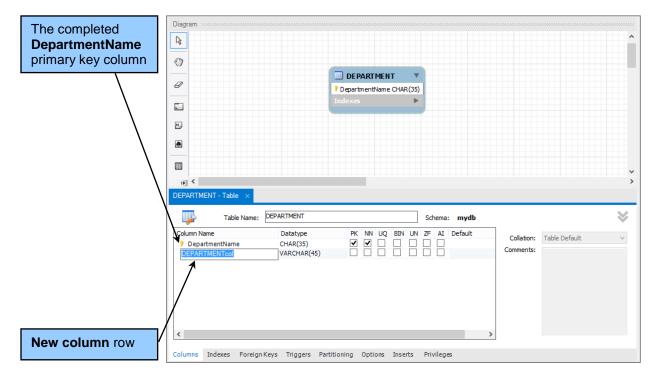


Figure E-12 — The Completed DepartmentName Primary Key Column

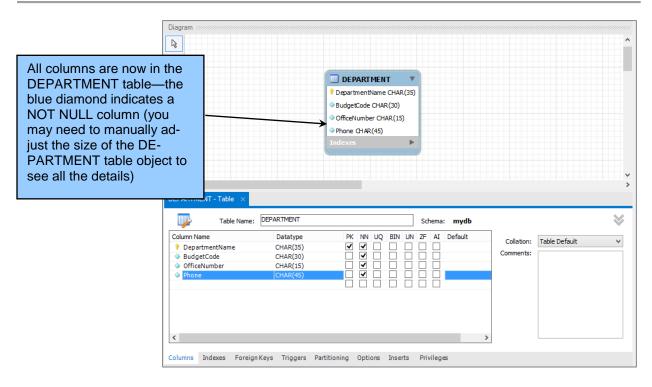


Figure E-13 — The Completed DEPARTMENT Table

- 6. Enter the other columns for the DEPARTMENT table, as shown in Figure E-3, into the DEPART-MENT table. Where Figure E-3 shows a data type of Text, you should use the Char data type, and where it shows that the columns is required, you should make it NOT NULL.
- 7. Click the **Save Model to Current File** button to save the changes, and then close the MySQL Table Editor by clicking the **X [Close]** button on the DEPARTMENT Table tab.

Now we will build the EMPLOYEE and PROJECT tables, using Figure E-3 as a guide, 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 E-14. 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 E-14.

Now that we have created the tables, we need to connect them with relationships. As shown in Figure E-15, 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 5), so the **1:1** identifying relationship does not make sense. Similarly, pure N:M relationshipsonly 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.

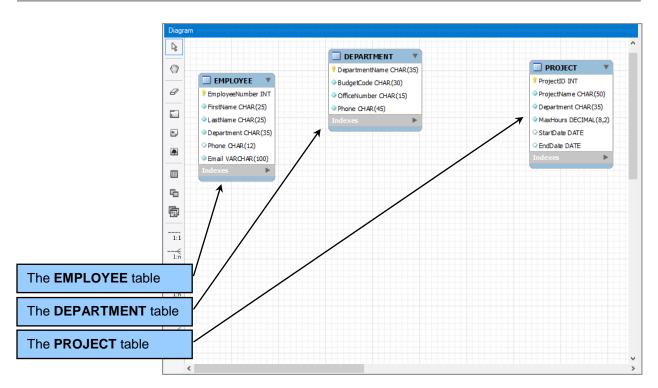


Figure E-14 — The Completed DEPARTMENT, EMPLOYEE and PROJECT Tables

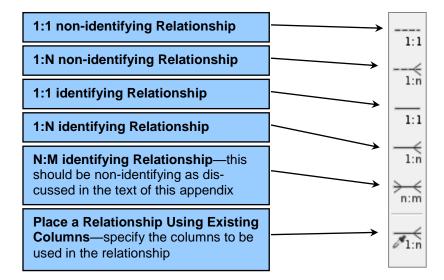


Figure E-15 — Relationships in the MySQL Workbench EER Diagram Toolbar

The MySQL Workbench uses the term **non-identifying** relationship, whereas in *Database Processing* 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 Processing* 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 E-16.
- 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 E-17.
- 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 E-18.
- 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 E-19. 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 E-20.

The MySQL Relationship Editor shown in Figure E-20 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.

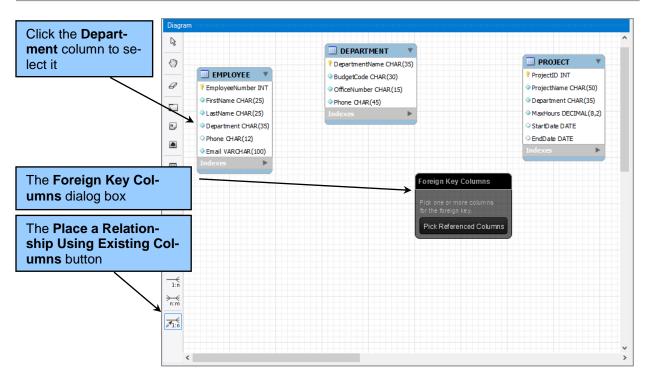


Figure E-16 — The Foreign Key Columns Dialog Box – Pick Columns for Foreign Key

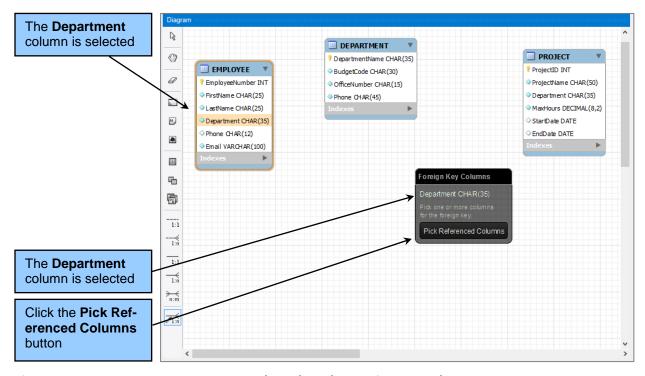


Figure E-17 — EMPLOYEE.Department Selected As The Foreign Key Column

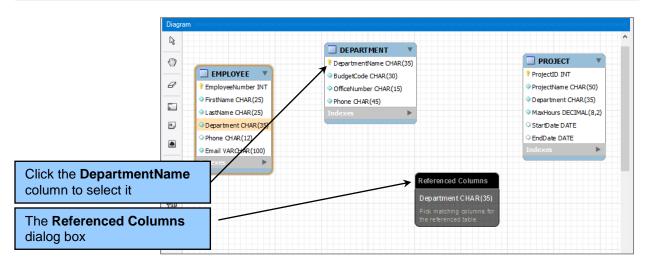


Figure E-18— The Referenced Columns Dialog Box – Pick Referenced Primary Key

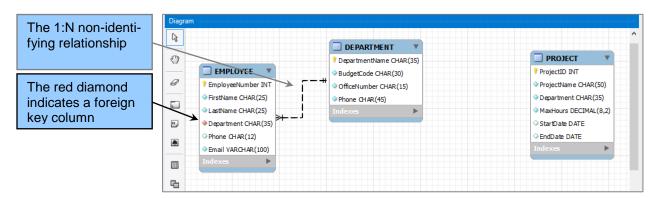


Figure E-19 — The Completed 1:N Non-identifying Relationship

The options on the Foreign Key page shown in Figure E-20 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 participartion 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.

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.

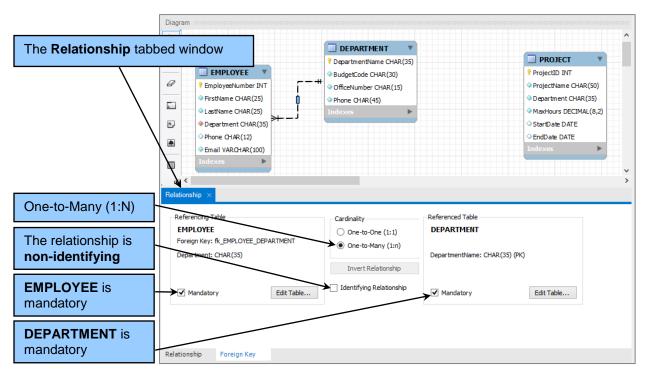


Figure E-20 — The Foreign Key Properties

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 E-21.

At this point we still need to build the ASSIGNMENT table and its relationships with EMPLOYEE and PROJECT. However, since, ASSIGNMENT is an association table in an associations relationship with EMPLOYEE and PROJECT (association relationships are discussed in Chapters 5 and 6), and since association tables are just intersection tables with additional attributes (again, as discussed in Chapters 5 and 6), we will use this as an opportunity to illustrate how MySQL Workbench handles N:M relationships.

As discussed in Chapters 5 and 6, 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 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.

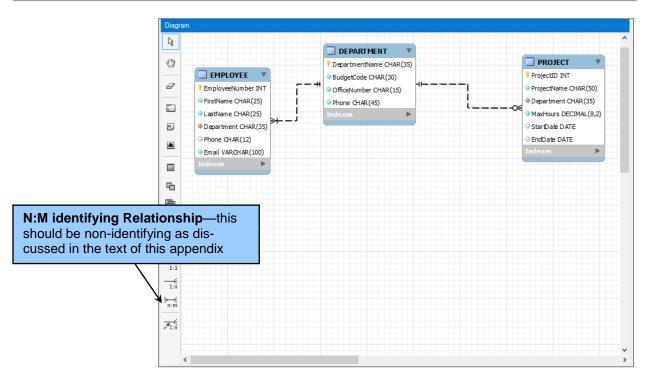


Figure E-21 — The Completed DEPARTMENT, EMPLOYEE and PROJECT Relationships

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 E-22, the MySQL Workbench creates an *intersection table* named EM-PLOYEE\_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) EM-PLOYEE\_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 an 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 Figure E-23.

- 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.

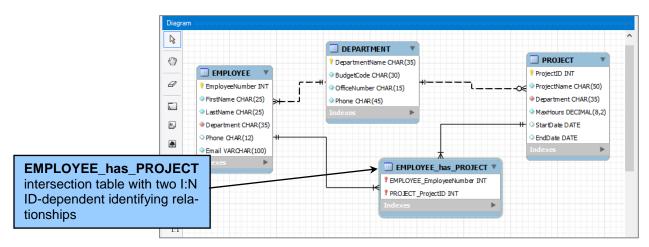


Figure E-22 — The Completed Set of Relationships

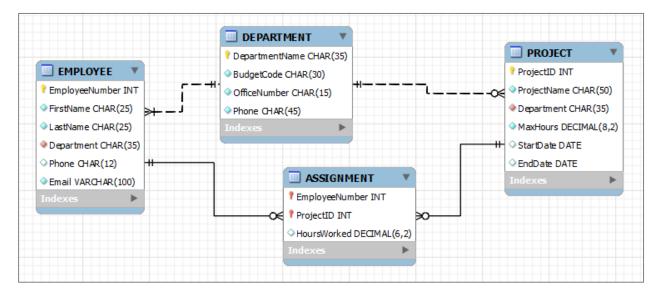


Figure E-23 — The Completed E-R Diagram

# 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	Result(1) window
button	
Root	Schema
SQL script	

# **Review Questions**

- E.1 What is MySQL?
- E.2 What is the MySQL Connector/ODBC?
- E.3 What is the MySQL Workbench?
- E.4 What are two purposes of the MySQL Workbench?
- E.5 The MySQL Workbench creates "database models". What is a data model? What is a database design? Which does MySQL Workbench create?
- E.6 How do you create a new E-R diagram in the MySQL Workbench?
- E.7 How do you create a table in an E-R diagram?
- E.8 How do you create a 1:N non-identifying relationship in an E-R diagram?
- E.9 How does the MySQL Workbench handle N:M relationships?

# **Project Questions**

- E.10 If you haven't already done so, download and install the MySQL Workbench.
- E.11 A database design for the Highline University database is shown in Figure 6-28. In the MySQL Workbench, create a new EER Model named HU-Database-Design. Recreate the database design shown in Figure 6-28 in this drawing.
- E.12 The database design for the VRG database is shown in Figure 6-39. In the MySQL Workbench, create a new EER Model drawing named VRG-Database-Design. Recreate the database design shown in Figure 6-39 in this drawing.