Database Processing

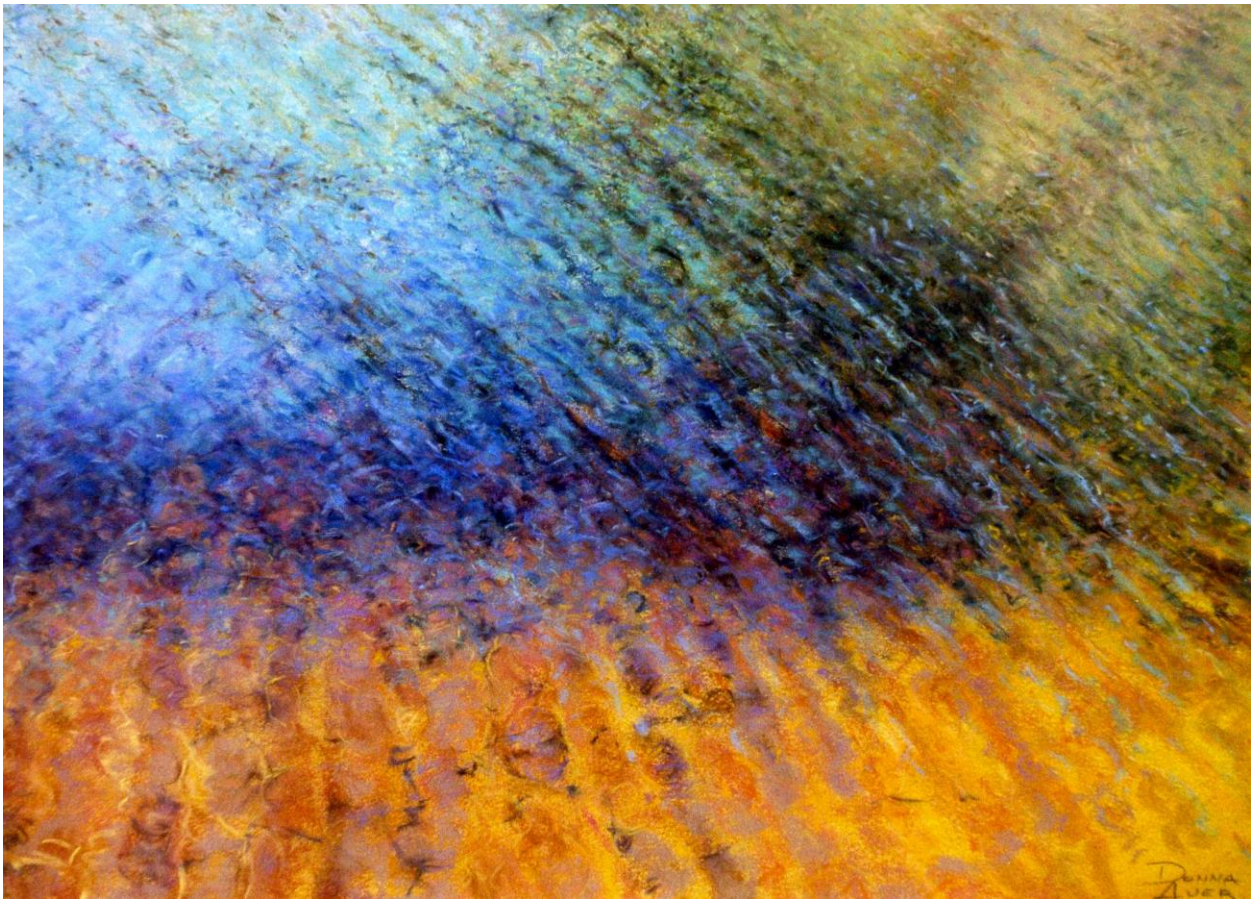
Fundamentals, Design, and Implementation

14th Edition

David M. Kroenke • David J. Auer

Online Appendix F

Getting Started with Microsoft Visio 2013



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 12*c* and Oracle Database Express Edition 11*g* 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.txt*. 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

PEARSON

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

Appendix F — 10 9 8 7 6 5 4 3 2 1

Chapter Objectives

- To learn how to create data models in Microsoft Visio 2013

What Is the Purpose of This Appendix?

Microsoft Visio 2013 is a component of Microsoft Office 2013, and is widely used to produce a variety of diagrams. There are three versions of Microsoft Visio 2013: Standard, Professional, and Premium. The templates that we use in this appendix are available in both the Professional and Premium editions.[1]

In this appendix, we are primarily interested in Microsoft Visio's ability to create data models (unfortunately, it cannot create database designs). We will also take a brief look at its ability to create process diagrams using Business Process Modeling Notation (BPMN).

Why Should I Learn to Use Microsoft Visio 2013?

For the purposes of this book, the most important reason to learn to use Microsoft Visio is that it provides a convenient method of creating the data models discussed in Chapter 5. Unfortunately, Microsoft Visio data models do not provide true linking between entities. Nonetheless, if you have Microsoft Visio 2013 Professional, you will have a tool capable of producing the data models and database designs required in the solutions to Review Questions and Project Questions in Chapter 5

[1] Students registered in CS or MIS classes should check to see if your school is part of the Microsoft DreamSpark program (https://www.dreamspark.com/). If so, you may be able to obtain Microsoft Visio 2013 Professional through that program.

What Will This Appendix Teach Me?

As its title implies, this appendix is designed to get you started using Microsoft Visio 2013 to create data models so that you have a tool to create these diagrams.

What Won't This Appendix Teach Me?

The material in this appendix does not go beyond what is necessary to get you started using Microsoft Visio 2013 on the topics covered in Database Processing (14th Edition) [hereinafter referred to as **DBP**]. There are many important Microsoft Visio topics not covered here, including organization charts, project management diagrams (such as Gantt charts and Pert charts), network diagrams (such as Active Directory tree), and Web site maps.

How Do I Start Microsoft Visio 2013?

To start Microsoft Visio 2013 running in Microsoft Windows 8.1, click the **Microsoft Visio 2013** tile on the Start screen as shown in Figure F-1 (For Windows 7, select **Start | All Programs | Microsoft Office | Microsoft Visio 2013**). The Microsoft Visio 2013 splash screen is displayed, followed by the Microsoft Visio 2013 window with the Home page displayed, as shown in Figure F-2.

Microsoft Visio 2013 uses the Microsoft Office Fluent user interface found in most (but not all) Microsoft Office 2007, Office 2010 and Office 2013 products. For more information on this interface, see Appendix A, pages A-8 – A-10. Here we note that Microsoft Visio 2013 opens with a variant of the **New tab** and associated page of the **Backstage view** displayed (the Backstage view is displayed when the **File command tab** is selected).

How Do I Create a Database Model Diagram in Microsoft Visio 2013?

To open a template for data models and database design, click the **Database** button to display the available software and database templates, as shown in Figure F-2. Click the **Crow's Foot Database Notation** template to select it, as shown in Figure F-3. The Crow's Foot Database Notation is the one that Microsoft Visio 2013 uses for data models with the IE Crow's Foot notation as used in Chapter 5. Note that the name *Database Notation* is somewhat misleading, as we are creating data models using E-R diagrams as we understand them from our discussion in Chapter 5. Nonetheless, this is the template we want, and we will have to sort things out as we go along. Also note that there are templates available for use with IDEF1X notation as discussed in Appendix C, and with UML notation as discussed in Appendix D. We briefly mentioned both the IDEF1X and UML notations in Chapter 5, but we are using IE Crow's Foot notation as our standard in this book. Again, we will just have to sort things out as we go along. Click the **Create** button to create a new Database Model Diagram document, as displayed in Figure F-5.

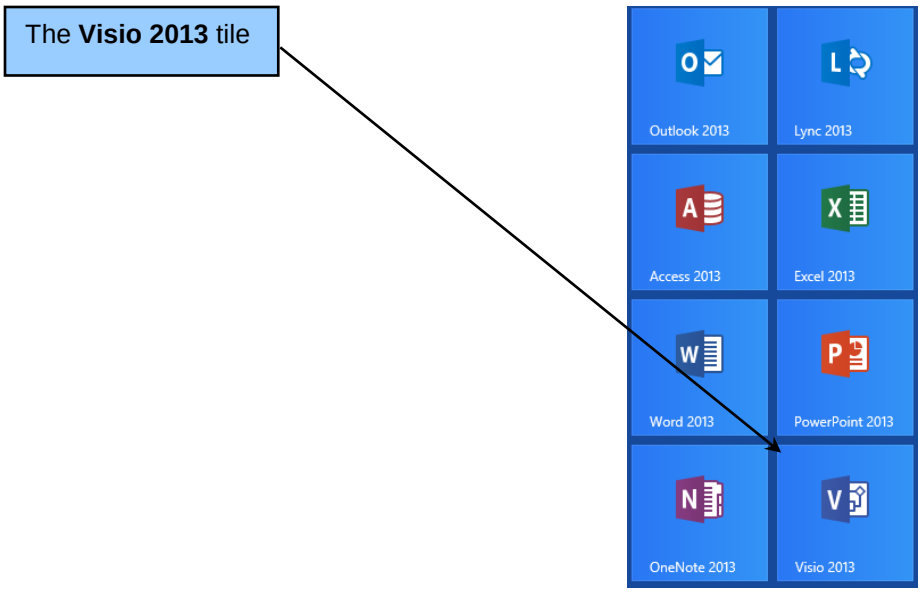


Figure F-1 — The Microsoft Visio 2013 Windows 8 Tile

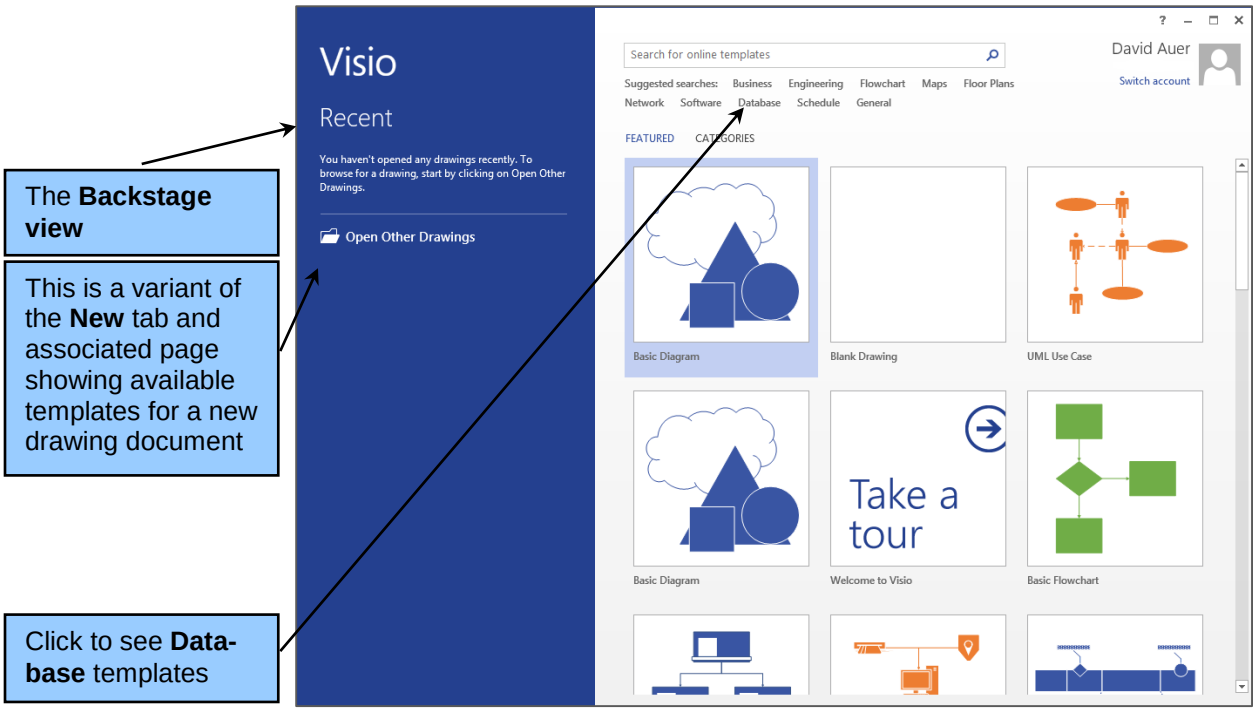


Figure F-2 — The Microsoft Visio 2013 Backstage View – Splash Screen

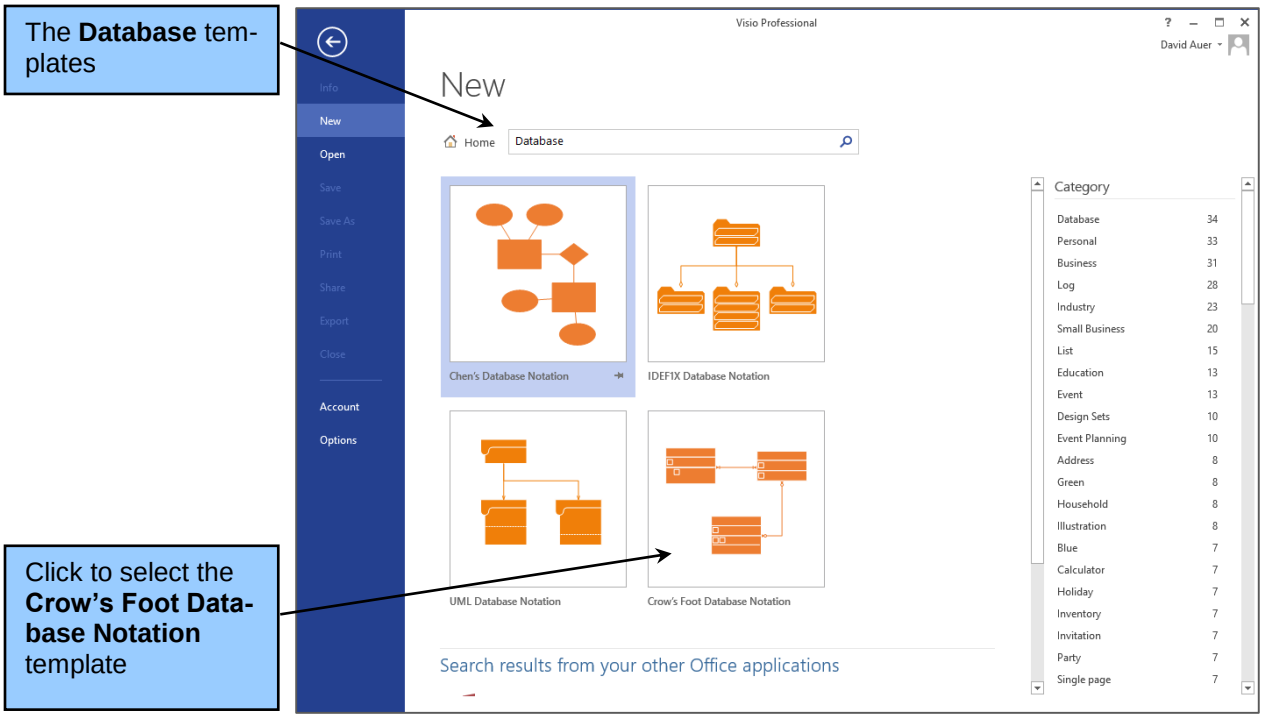


Figure F-3 — The Database Templates

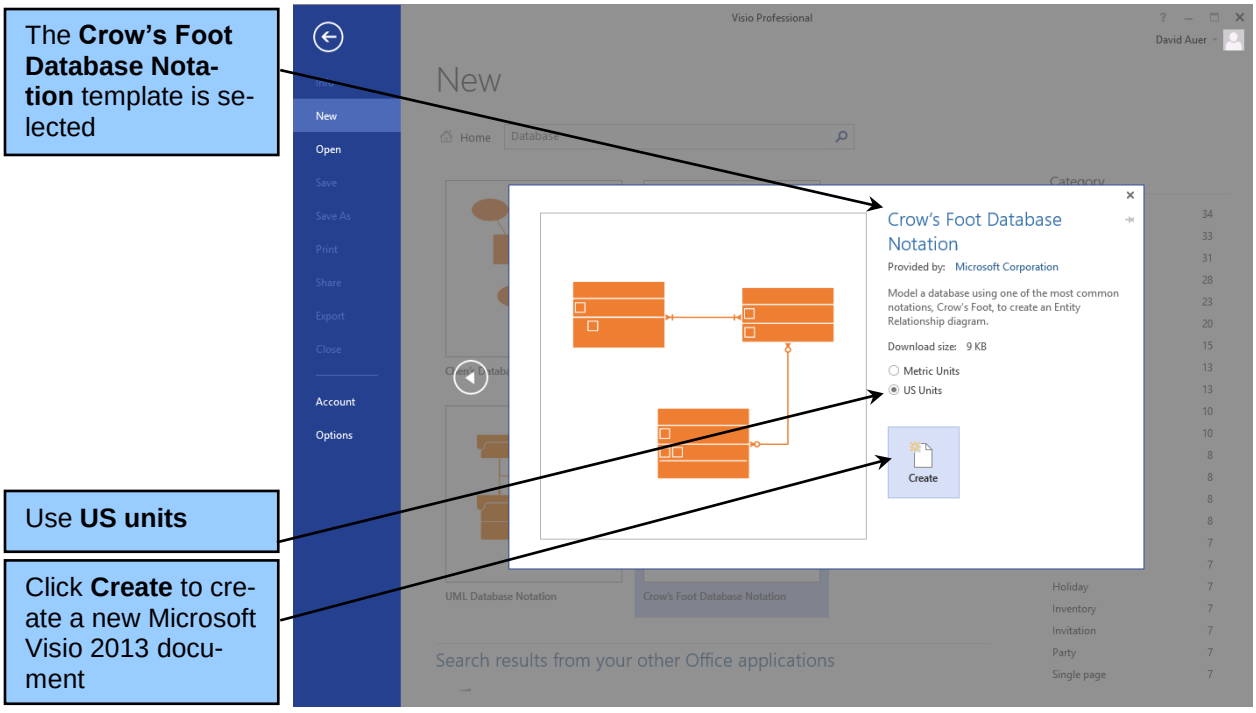


Figure F-4 — The Crow's Foot Database Notation Template

As shown in Figure F-5, a new, blank diagram document, named Drawing1, is created, the appropriate Shapes stencils (a **stencil** is Microsoft Visio 2013's term for a group of template objects) are displayed in the Shapes window. Figure F-6 shows more detail of the Shapes window, including:

- The **Minimize the Shapes window** button — As the name says, use this to minimize the Shapes window.
- The **More Shapes** button — Use this to display the **More Shapes list**, as shown in Figure F-6, if you need to add additional shape templates to the drawing.
- The **Crow's Foot Database Notation** button — Use this to display the Entity Relationship (US units) shapes pane.
- The **Crow's Foot Database Notation** stencil — The set of shape objects in the Entity Relations (US units) stencil.

Figure F-7 shows the same screen, but illustrates what happens when we click on one of the command tabs. Here, we have clicked on the HOME tab, and the associated ribbon with the HOME tab command groups is displayed. This is how we access commands on the ribbon, but note that when we do this we cover up some of the drawing we are working on.

The Crow's Foot Database Notation stencil contains all the template objects we will use to build our data models. As shown in Figure F-8, it contains:

- The **Entity** object — Use this object to create *entities* for data models in the Microsoft Visio database model diagram.
- The **Relationship** object — Use this object to create relationships between *entities* in *data models* (it is actually what Microsoft usually refers to as a dynamic connector).

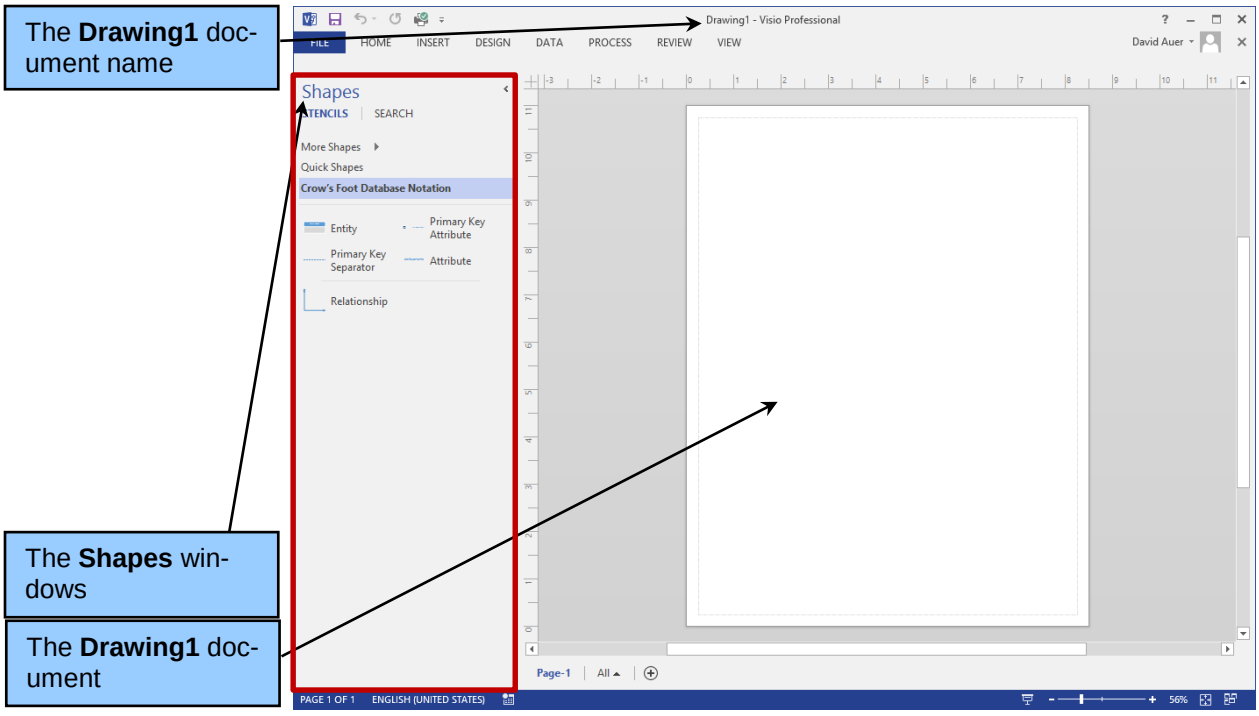


Figure F-5 — The Microsoft Visio 2013 Data Model Diagram

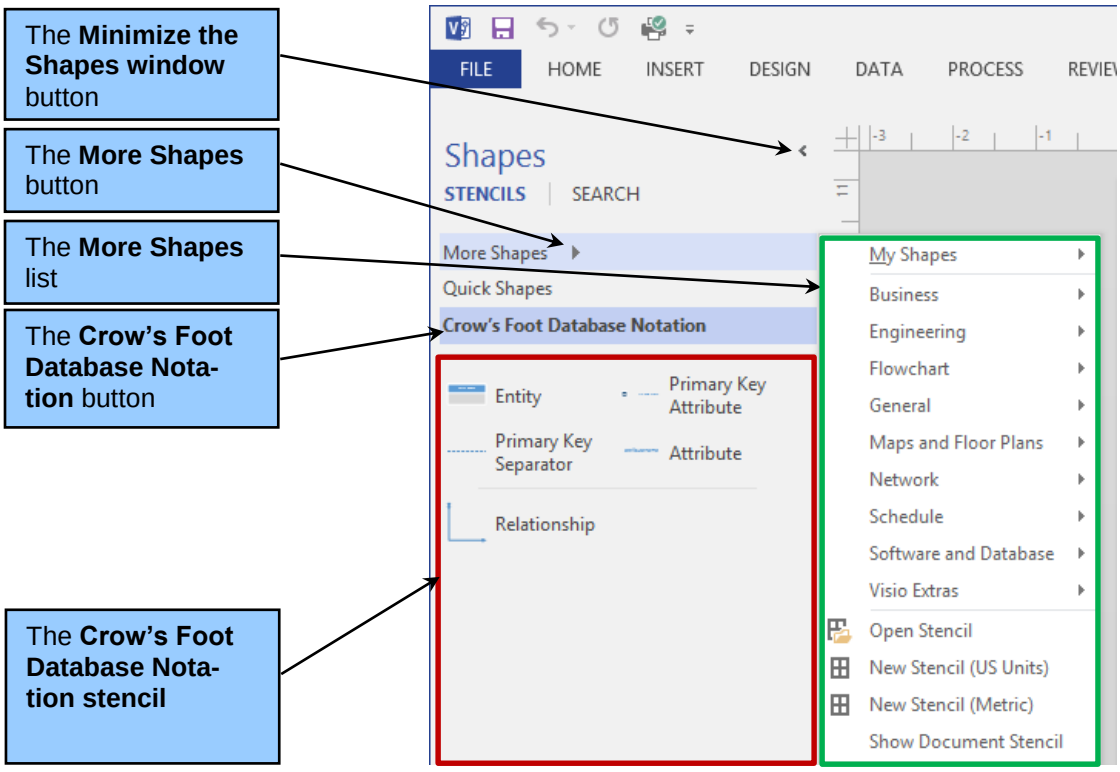


Figure F-6 — The Shapes Window

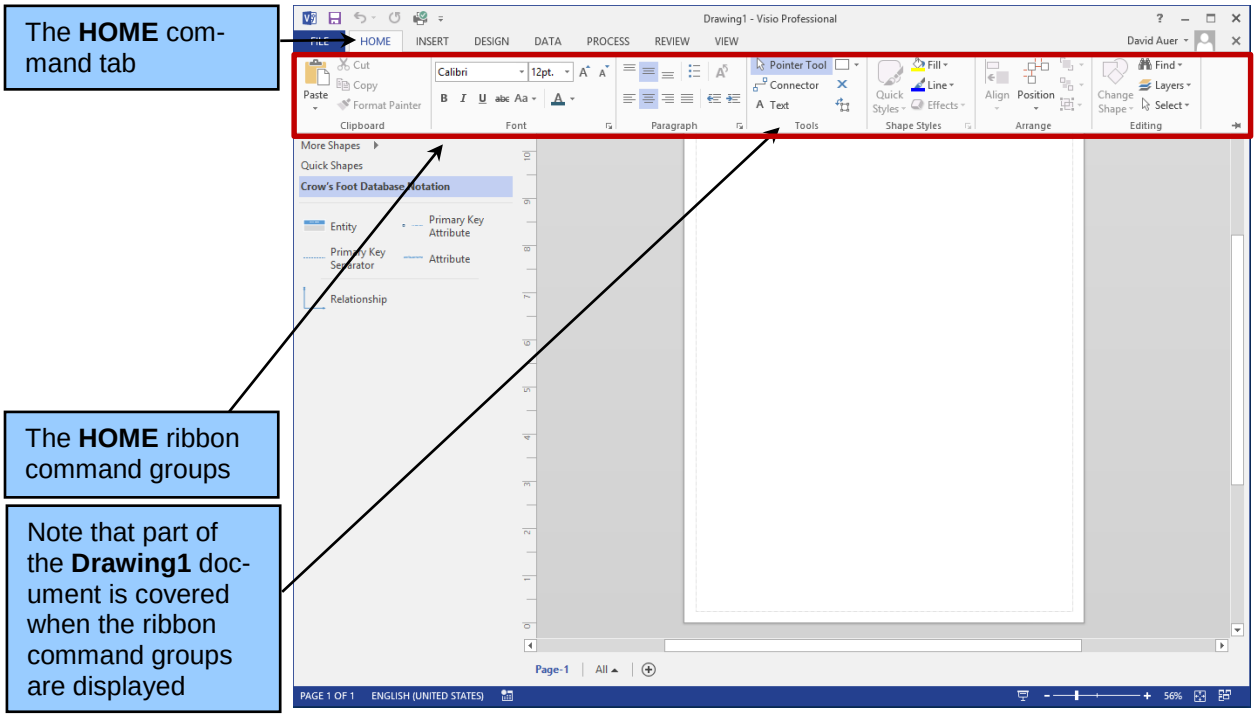


Figure F-7 — The HOME Command Tab Ribbon

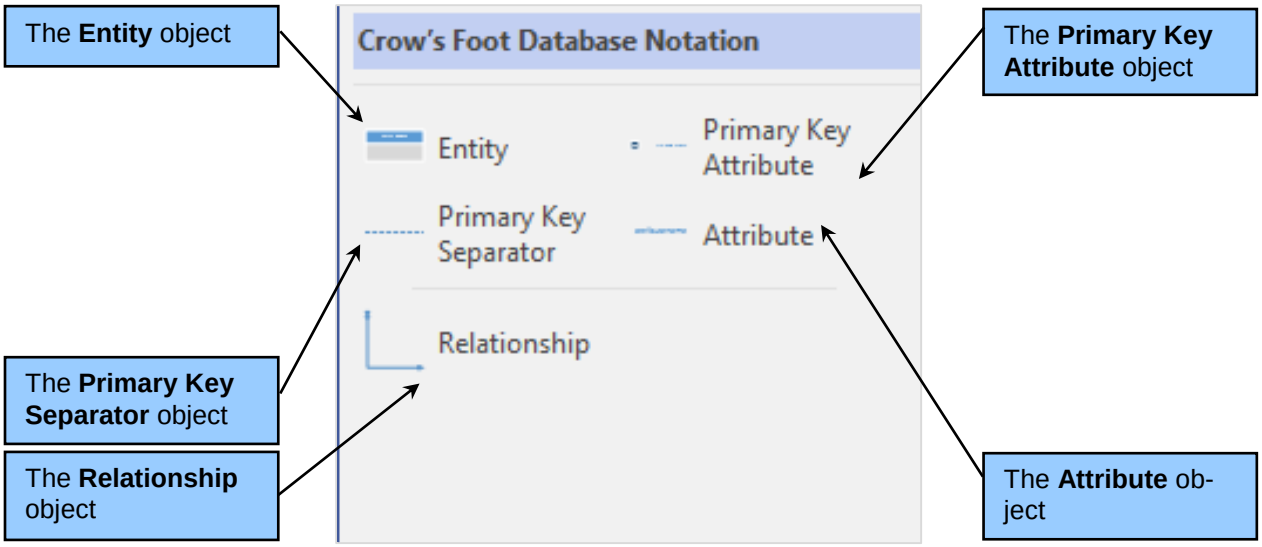


Figure F-8 — The Crow's Foot Database Notation (US units) Stencil Objects

- The **Primary Key Attribute** object — Use this object to add additional primary key attributes (as, for example, in composite primary keys) to the *entities* in data models.
- The **Primary Key Separator** object — Use this object to insert a visible line between primary key attributes and non-key attributes in the *entities* in data models.
- The **Attribute** object — Use this object to add additional attributes to *entities* in data models.

Figure F-9 shows the Microsoft Visio 2013 window with:

- The **Shapes** windows minimized — Note that all of the Entity Relationship (US units) stencil objects are still visible.
- The **Expand the Shapes window** button — Use this to restore the Shapes window to its full size.
- The **Quick Access Toolbar** — Added **Quick Print** and **Print Preview** buttons are displayed.The Quick Access Toolbar, and which buttons are displayed on it, is discussed in Appendix A.
- The **Zoom** controls — Use these to control the displayed size of the drawing, now shown at 56%.

By configuring the Microsoft Visio 2013 window in this manner, we have a much more efficient working area.

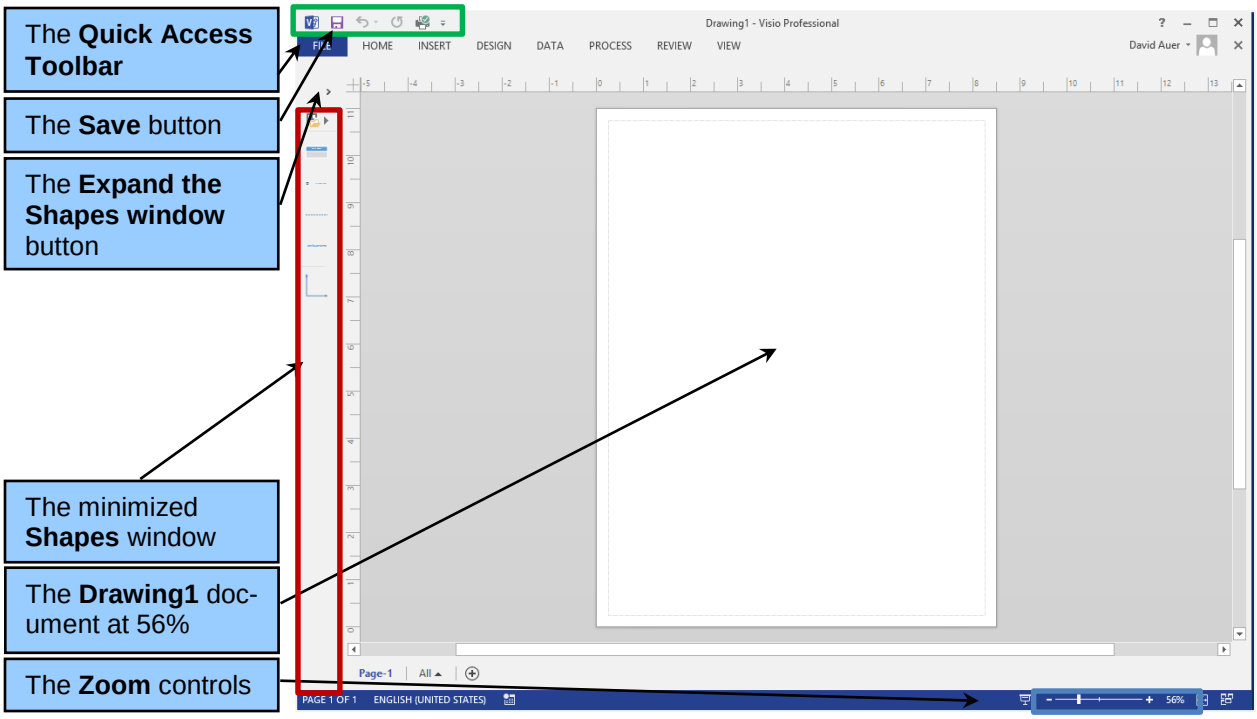


Figure F-9 — The Microsoft Visio 2013 Database Model Diagram with Minimized Shapes Window

How Do I Name and Save a Database Model Diagram in Microsoft Visio 2013?

We will illustrate creating data models and database designs in Microsoft Visio 2013 by creating an E-R diagram for the Wedgewood Pacific Corporation (WPC) database that we created in Microsoft Access as part of the Project Questions in Chapters 1 and 2, and in SQL Server 2014 in Chapter 2.[2] The column characteristics for each of the the WPC tables are shown in Figure F-10.

To save a Microsoft Visio 2013 database model diagram drawing, click the Save button in the Quick Access Toolbar. The first time we save the drawing, Microsoft Visio 2013 opens a **Save As** dialog box. This is a standard Microsoft Office 2013 dialog box. Name the drawing as **WPC-Data-Model**.

[2] Of course, it could be agrued 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	Type	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	Type	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	Type	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	Type	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 F-10 — The WPC Database Table Column Characteristics

As we discuss in Chapters 5 and 6, 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. So, we will build the entities we need, and then consider exactly how Microsoft Visio 2013 handles data models.

Also as we discuss in Chapters 5 and 6, the diagrams we are creating are entity-relationship models, and we are using the IE Crow's Foot notation as our standard. We will also see how well Microsoft Visio 2013 supports IE Crow's Foot notation.

How Do I Create Entities in a Database Model Diagram in Microsoft Visio 2013?

Now that we have a blank document to use as an E-R diagram work area available, we can build the E-R diagram itself. We start by adding an entity to the E-R diagram. We will add the DEPARTMENT entity. By looking at the column characteristics of the DEPARTMENT table in Figure F-10, we can see the columns that are used in the DEPARTMENT table.

Creating an Entity in the Microsoft Visio E-R Diagram:

1. In the Shapes window, click-and-hold the **Entity** object shown in Figure F-8.
2. Move the cursor over the blank E-R diagram area, and then release the left mouse button. A new entity object named Entity Name is created on the E-R Diagram, as shown in Figure F-11.
3. Each line of text in the entity is a formatted text box. To edit the text in a text box, just double-click the text.

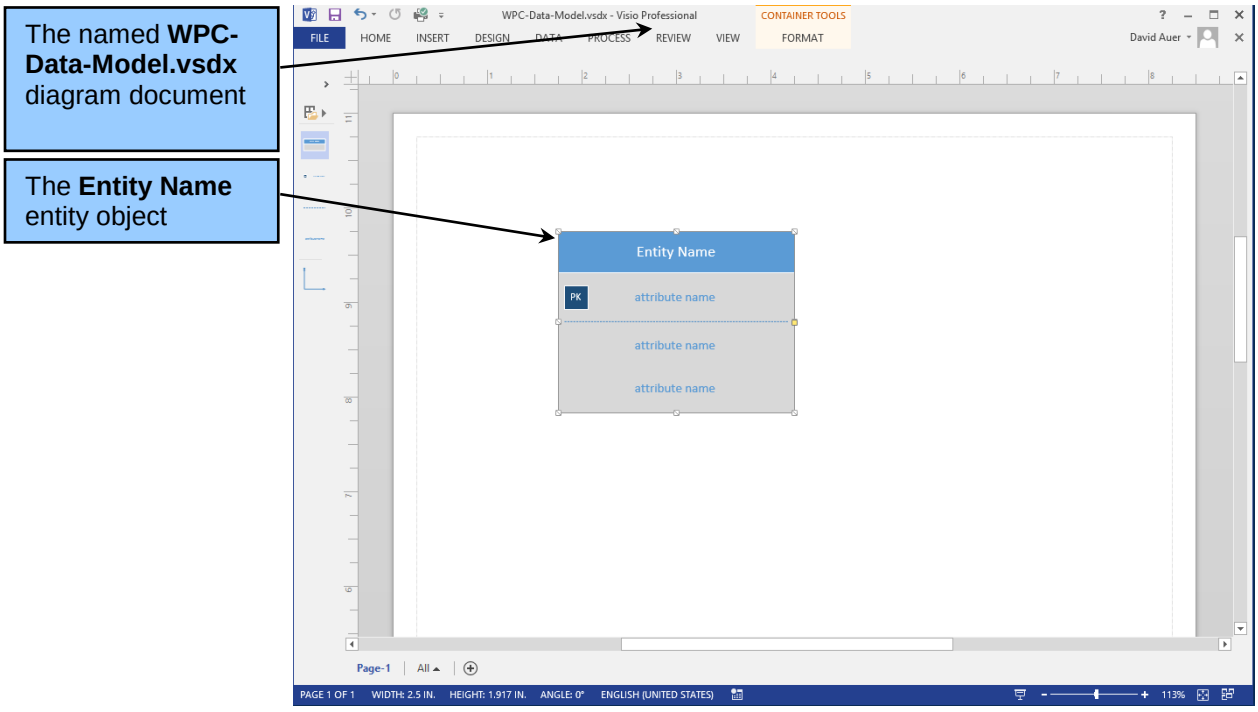


Figure F-11— The Entity Name Entity Object

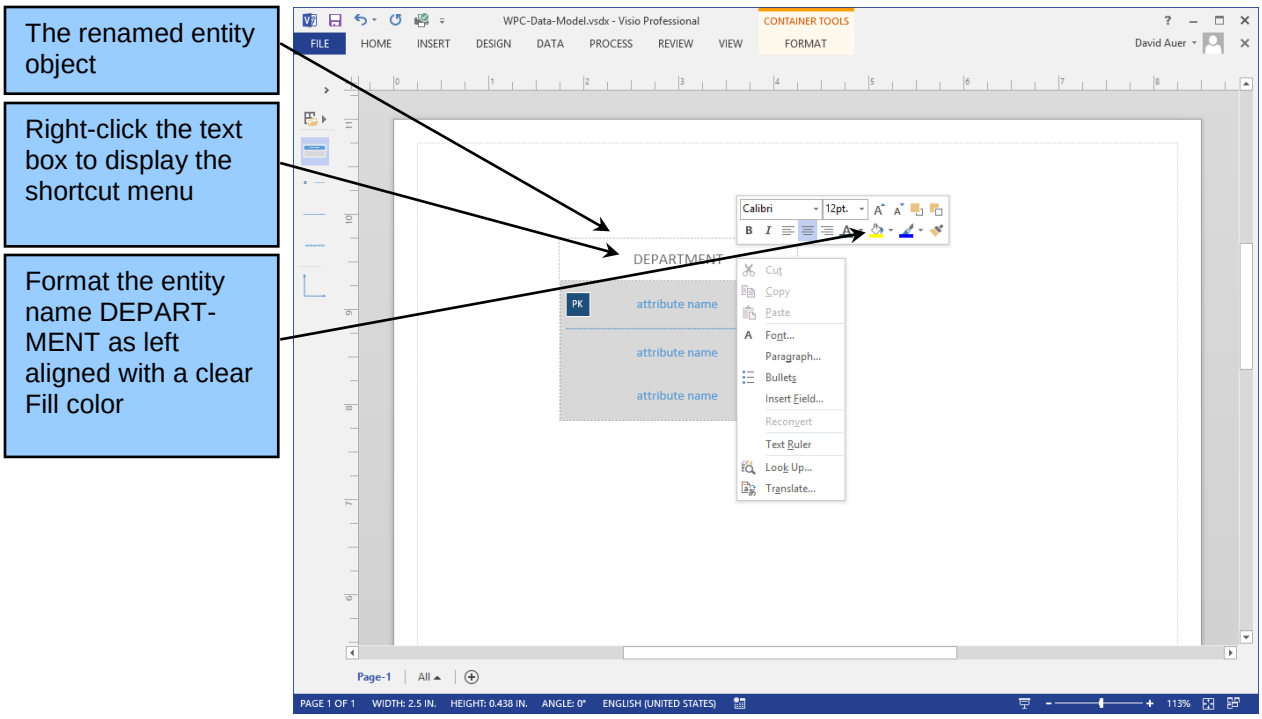


Figure F-12 — Naming and Formatting the Entity

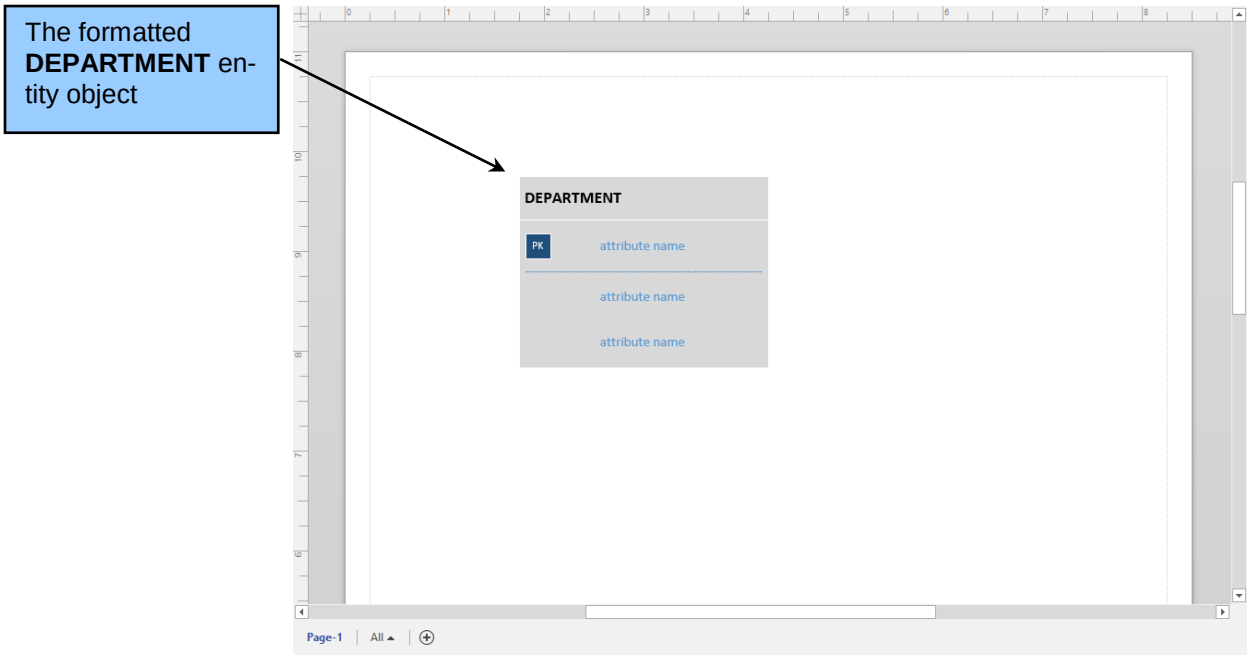


Figure F-13 — The Named and Formatted DEPARTMENT Entity

4. Double-click Entity Name to select it, and type in the table name **DEPARTMENT**.
5. Right-click the text to display the shortcut menu as shown in Figure F-12.
6. Format the table name DEPARTMENT as 12pt, bold, black text with no fill color. The final entity name appears as shown in Figure F-13.

Now that the we have named the DEPARTMENT entity, we have to deal with the fact that data models and the entities in them do not have primary keys—they have identifiers. There is no symbol, such as PK for primary key, for an identifier. Therefore, we have to make a choice—either use the PK text boxes for identifiers, or not use these boxes at all.

For the sake of simplicity, we will use the PK boxes for our identifiers. This simply cuts down on the work we have to do to create the data models. But as we gain more skill using Visio 2013, we opt for more work and a cleaner and more consistent data model approach.

Also notice that what we are calling columns in Figure F-10 are being called attributes in the Visio shapes. This is okay, because the two terms are synonyms, and the term attributes is correct for data models.

Creating the DEPARTMENT Entity Identifier (PK) Attribute in the Visio E-R Diagram:

1. In the DEPARTMENT entity object, double-click the **PK attribute name** text box.
2. Edit the text to read **DepartmentName**.
3. Format the text as left aligned, normal, black text with no fill.
4. The DEPARTMENT entity object now appears as shown in Figure F-14.
5. The primary key column name for DEPARTMENT (as shown in the column characteristics Key listings in Figure F-10) is DepartmentName. Click the first *attribute name* text box, and type the column name **DepartmentName**. The diagram now appears as shown in Figure F-15.

Naming the other attributes is done the same way. And if we need more attributes (which we will for the *Phone* attribute), we just drag a new attribute object and attach it to the bottom of the entity object. The final DEPARTMENT entry is shown in Figure F-15.

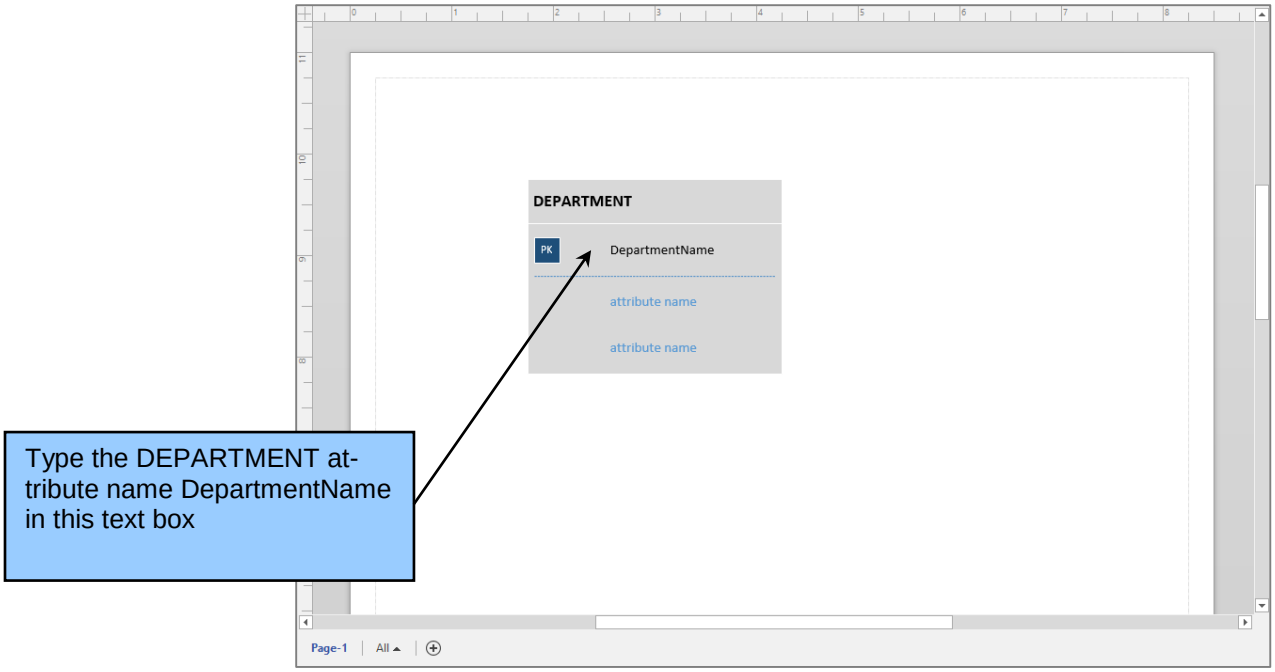


Figure F-14 — Naming the Primary Key Attribute

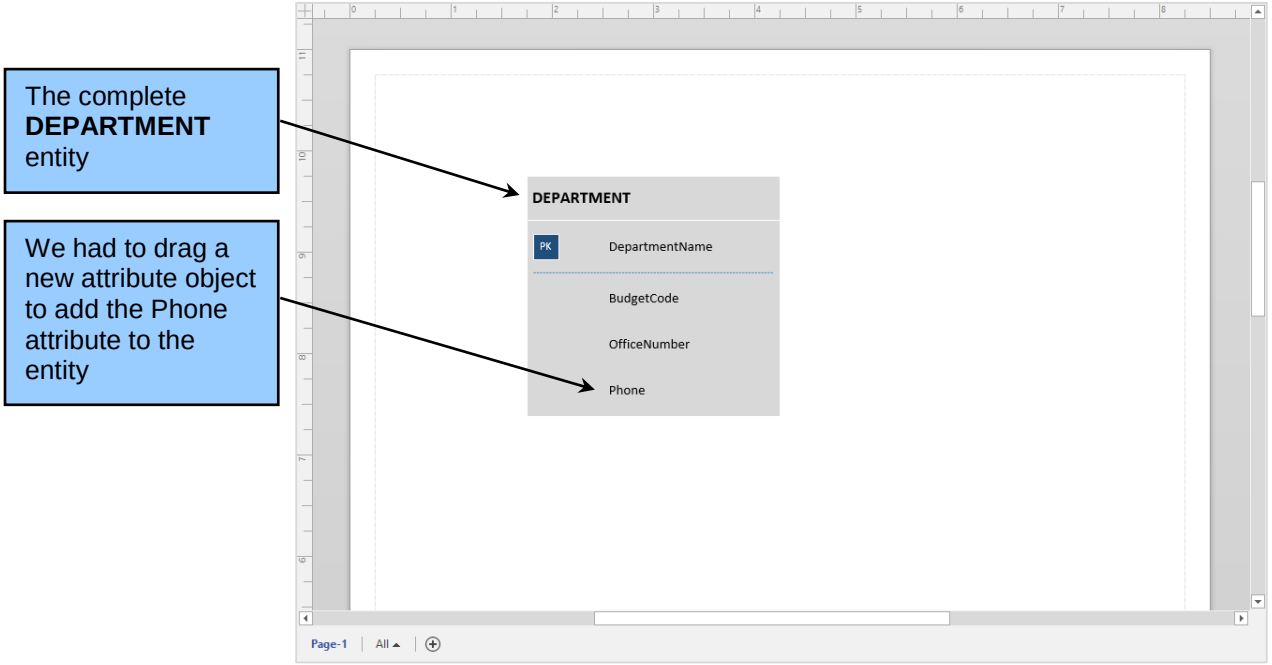


Figure F-15 — The Complete DEPARTMENT Entity

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 F-16. Note that the table objects have been resized and rearranged, and the page itself is in landscape orientation instead of portrait orientation.

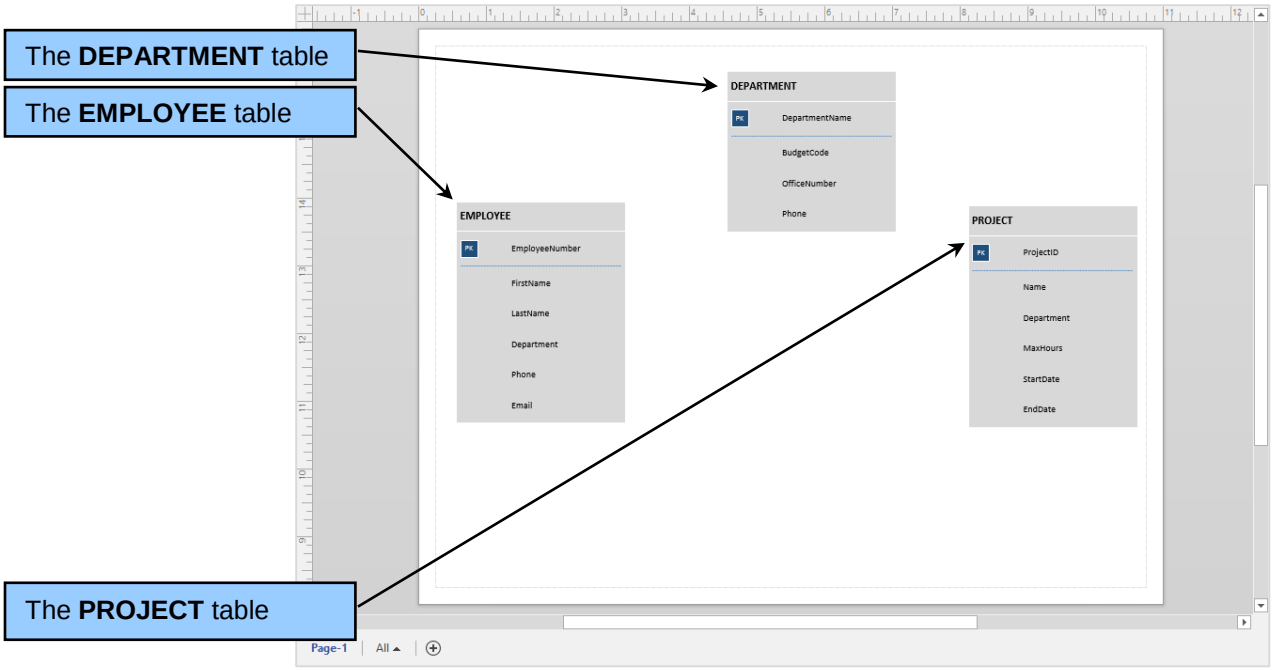


Figure F-16 — The Completed DEPARTMENT, EMPLOYEE and PROJECT Tables

How Do I Create Relationships Between Entities in a Data Model Diagram in Microsoft Visio 2013?

Now that we have created the entities, we need to connect them with relationships. To do this we use the **Relationship** connector. This is really a renamed form of the standard Microsoft Visio 2013 **Dynamic connector**—used between two entities, this is just a line with formattable line endings.

The type of relationship displayed and other relationship parameters is controlled by the connector properties. That is how we know if we have, for example, a **1:1 non-identifying**, **1:N non-identifying**, or **1:N identifying** relationships.

Microsoft Visio 2013 uses the term non-identifying relationship, whereas in *DBP* 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 Microsoft Visio 2013 uses *non-identifying*, we will also use that term in this appendix for consistency with the Microsoft Visio 2013 screen shots, while remaining well aware that we have used *nonidentifying* in *DBP* itself.

It is also how we handle N:M relationships. 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.

Now we will build a data model and a database design to consider exactly how Microsoft Visio 2013 handles these diagrams.

How Do I Create Data Models Using Relationships in Microsoft Visio 2013?

At this point, we need to create two 1:N non-identifying relationships, one between DEPARTMENT and EMPLOYEE and one between DEPARTMENT and PROJECT. We will start with the relationship between DEPARTMENT and EMPLOYEE

Creating a 1:N Nonidentifying Relationship Between Two Tables:

1. Click-and-hold the **Relationship** object.
2. Drag the object into the E-R diagram, and move it towards the center of the EMPLOYEE table. When the EMPLOYEE table is outlined in green, the connector is attached to the table object—release the mouse button. The E-R diagram now appears as shown in Figure F-17.
3. Drag the dynamic connector object in the E-R diagram, and drag the free end of it towards the center of the DEPARTMENT table. When the DEPARTMENT table is outlined in green, the connector is attached to the table object—release the mouse button. The E-R diagram now appears as shown in Figure F-18.
4. The term Dynamic Connector means that if we move any of the objects to which the connector is attached, then the connector will automatically adjust its size and position to match. In Figure F-19, we have moved the DEPARTMENT table towards the center of the E-R diagram, and the connector has adjusted itself as needed.

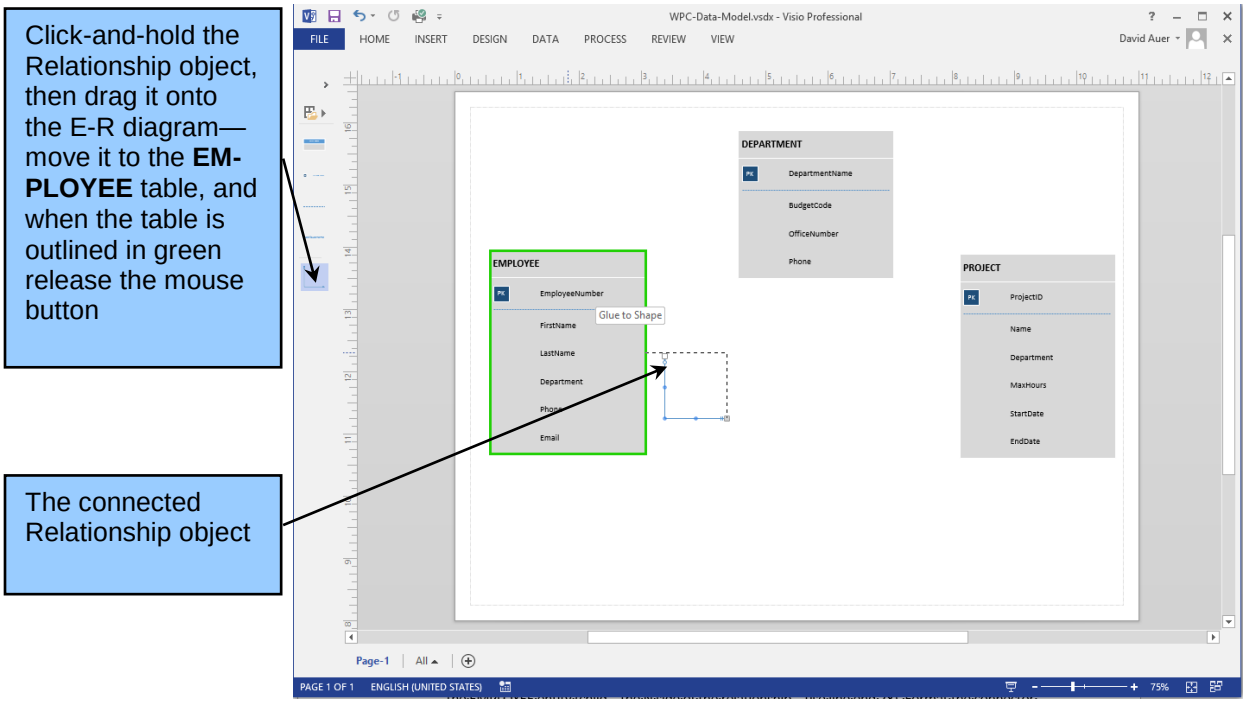


Figure F-17 — The Dynamic Connector Attached to the EMPLOYEE Table

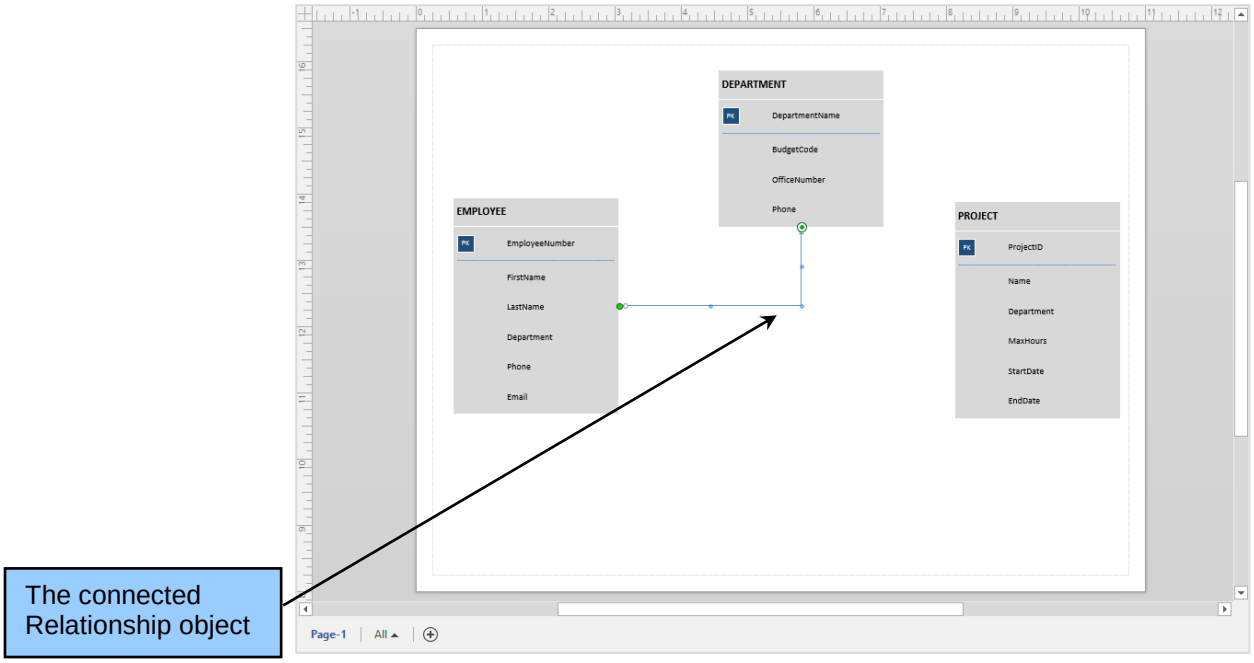


Figure F-18 — The Relationship Object Attached to Both the EMPLOYEE and DEPARTMENT Tables

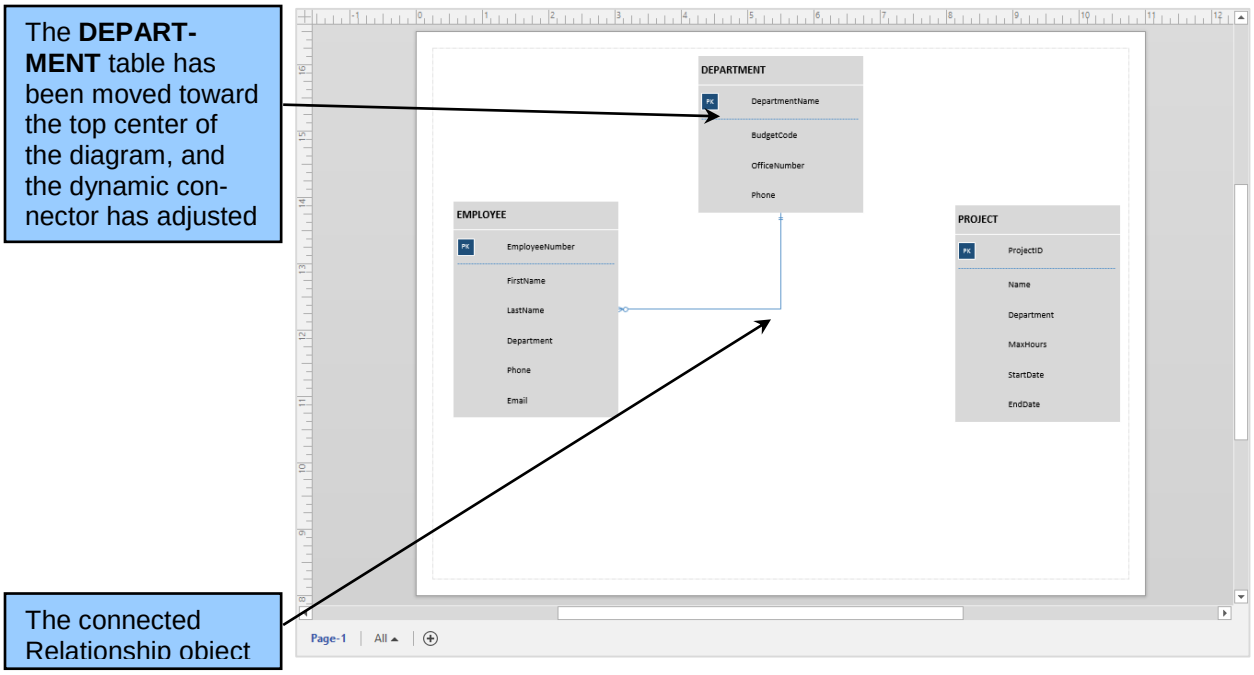


Figure F-19 — The Dynamic Connector Adjusts as DEPARTMENT Table is Moved

5. We can format the line properties of the relationship. Click the relationship object to select it, and then right-click to display the short-cut menu as shown in Figure F-20.
6. In the short-cut menu, click the **Set Begin Symbol** command to display the Set Begin Symbol menu. In the the Set Begin Symbol menu, note that Zero or more is set. To understand what Crow's Foot symbols this corresponds to, look at Figure F-21, which reproduces Figure 5-8.

"Zero or more" corresponds to the optional-many Crow's Foot symbol.

7. This is the correct setting for this line end, but note that we can easily adjust it to any of the four Crow's Foot symbols as needed.

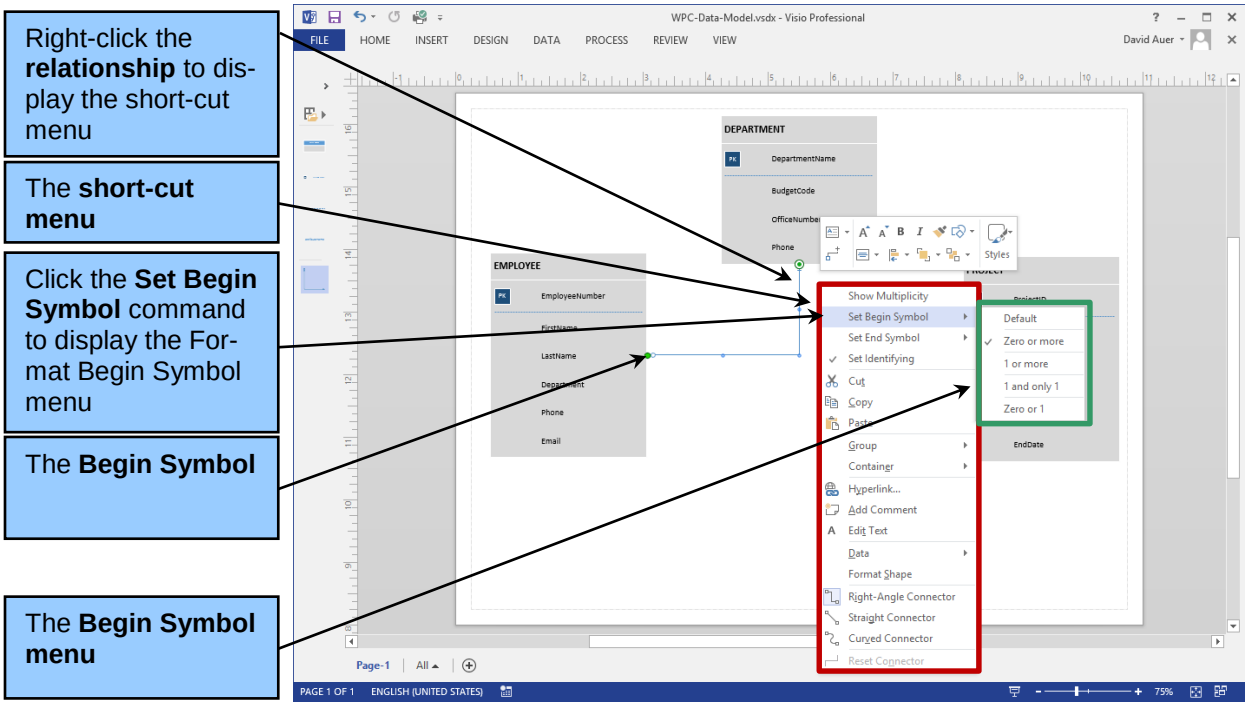


Figure F-20 — The Short-Cut Menu

Symbol	Meaning	Numeric Meaning
	Mandatory—One	Exactly one
	Mandatory—Many	One or more
	Optional—One	Zero or one
	Optional—Many	Zero or more

Figure F-21 — Crow's Foot Symbols and Line Symbols

8. Since this is a non-identifying relationship, the line should be a dashed line, but Microsoft Visio 2013 does *not* allow us to format the line as a dashed line (the formatting options are available, but the selected formatting is *not* applied to the line!).
9. The other end of the line is correct.
10. Click the **Save** button on the Quick Access Toolbar to save the work on the data model.

This allows us to format the relationship lines between the entities using the appropriate symbols for the style of E-R diagram we are creating. We can use these to create IE Crow's Foot data models. At this point, you need to simply experiment with how to format the connector lines in Microsoft Visio 2013.

The relationship between DEPARTMENT and PROJECT is also a 1:N non-identifying relationship, and can be created exactly the same way. You should create that relationship now, and save the data model work again.

Using Microsoft Visio 2013 relationships, we *can*, in fact, model N:M relationships. At this point, based on the WPC database in Figure F-10, we still need to build the ASSIGNMENT table and its relationships with EMPLOYEE and PROJECT. However, for our WPC data model, let's assume that we simply want an N:M relationship between EMPLOYEE and PROJECT. That is, employees will work on projects, and projects must have employees to work on them. Figure F-22 shows the N:M relationship, using the same dynamic connector construction.

The important question for a *data model* design tool is whether or not we can model an N:M non-identifying relationship between two entities. 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 enties). In a database

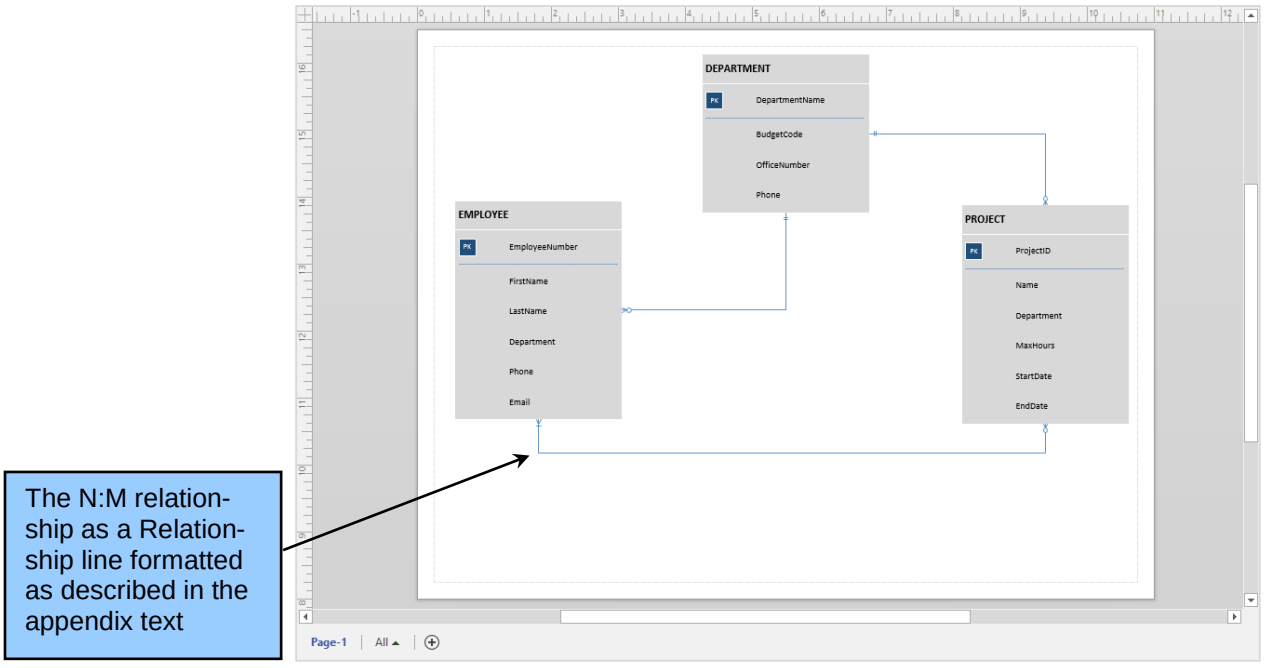


Figure F-22 — The N:M Relationship in the Data Model

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.

At this point, you should create the N:M relationship, and then save the WPC-Data-Model.vsdx file. We will leave adding the ASSIGNMENT table to the data model for the Exercises at the end of this appendix.

In this model, the non-identifying relationships are correct (all three entities are strong entities), but what about the minimum cardinalities? All relationships are shown as having mandatory 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.

We will also 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 for the N:M relationship are correct as created, and our relationship is correct as drawn. Note that if optional minimum cardinalites are allowed, we just need to change which line end we use!

Save the WPC-Data-Model.vsdx file a final time, and then click **File | Close** to close the drawing.

How Do I Create Database Designs Using Microsoft Visio 2013?

The unfortunate answer is you cannot do this well in Microsoft Visio 2013. Microsoft Visio 2010 could do this, and in our opinion was a much more flexible and versatile data modeling and database design tool than Microsoft Visio 2013. You can still open database designs created in Visio 2010, but without the previous functionality that they had in Visio 2010. They are simply unalterable block diagrams, as shown in Figure F-23. Why did Microsoft remove so much capability from Visio 2013? We have no idea. But what a shame that they did!

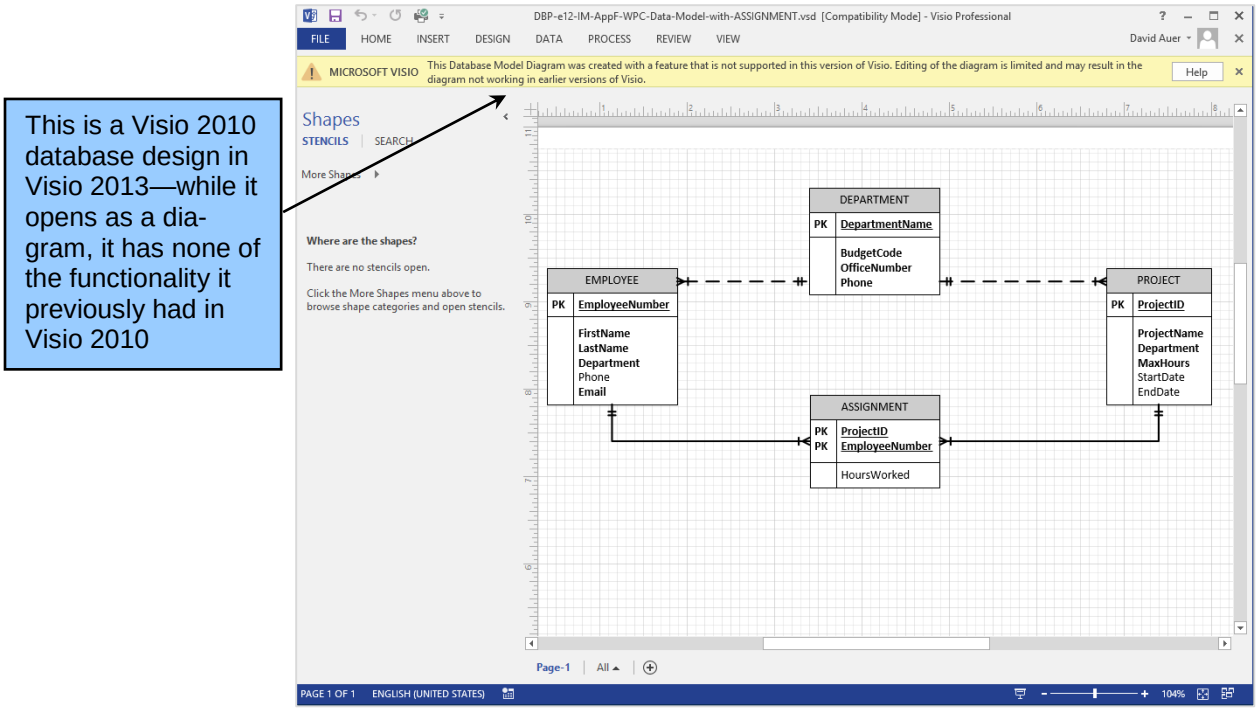


Figure F-23 — The Visio 2010 Database Design in Visio 2013.

Key Terms

1:1 non-identifying relationship	**1:N identifying relationship**
1:N non-identifying relationship	**Backstage view**
Crow's Foot Database Notation	**Databases button**
Entity object	**File command tab**
Microsoft Visio 2010	**Microsoft Visio 2013**
Minimize the Shapes window button	**More Shapes button**
More Shapes list	**Relationship object**
Stencil	

Review Questions

F.1 What is Microsoft Visio 2013?

F.2 What is the Backstage view?

F.3 How do you create a new database model diagram drawing in Microsoft Visio 2013?

F.4 Describe the Shapes window, and the objects in the Shapes window.

F.5 What shapes are included in the **Crow's Foot Database Notation** stencil, and what is the purpose of each shape?

F.6 How do you create entities in a database model diagram?

F.7 When should you use the Relationship connector object?

F.8 How do you format a Relationship object so that it displays IE Crow's Foot E-R notation?

F.9 How do you connect a Relationship object to two tables? What significance is there to which end of the Relationship is attached to which table?

F.10 How are Relationship connector cardinalities set? What is the correspondence between the cardinality terms used in the dynamic connector and the IE Crow's Foot E-R notation shown in Figure F-21?

Project Questions

F.11 If you haven't already done so, work through the steps described in this appendix to create the data model and database design for the WPC database.

F.12 Create a copy of the WPC-Data-Model.vsdx file, and name it WPC-Data-Model-with-ASSIGNMENT.vsdx. Modify the data model to include the ASSIGNMENT table as shown in the column characteristics in Figure F-10.

F.13 A data model for the Highline University database is shown in Figure 5-52. In Visio 2013, create a file named HU-Data-Model.vsdx, an use it to recreate the data model shown in Figure 5-52.

F.14 The data model for the View Ridgel Gallery VRG database is shown in Figure 6-37. In Visio 2013, create a file named VRG-Data-Model.vsdx, an use it to recreate the data model shown in Figure 6-37.