
Database Concepts

8th Edition

David M. Kroenke • David J. Auer

Scott L. Vandenberg • Robert C. Yoder

Online Appendix A

Getting Started with Microsoft SQL Server 2016



VP Editorial Director: Andrew Gilfillan
Senior Portfolio Manager: Samantha Lewis
Content Development Team Lead: Laura Burgess
Program Monitor: Ann Pulido/SPi Global
Editorial Assistant: Madeline Houp
Product Marketing Manager: Kaylee Carlson
Project Manager: Katrina Ostler/Cenveo® Publisher Services
Text Designer: Cenveo® Publisher Services

Interior design: Stock-Asso/Shutterstock; Faysal Shutterstock
Cover Designer: Brian Malloy/Cenveo® Publisher Services
Cover Art: Artwork by Donna R. Auer
Full-Service Project Management: Cenveo® Publisher Services
Composition: Cenveo® Publisher Services
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 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 XE 2016 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. This book is not sponsored or endorsed by or affiliated with The PHP Group.

Copyright © 2017, 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- author. | Auer, David J., author.
Database concepts / David M. Kroenke, David J. Auer, Western Washington University, Scott L. Vandenberg, Siena College, Robert C. Yoder, Siena College.
Eighth edition. | Hoboken, New Jersey : Pearson, [2017] | Includes index.
LCCN 2016048321 | ISBN 013460153X | ISBN 9780134601533
LCSH: Database management. | Relational databases.
LCC QA76.9.D3 K736 2017 | DDC 005.74--dc23
LC record available at <https://lccn.loc.gov/2016048321>

Appendix A — 10 9 8 7 6 5 4 3 2 1



Appendix Objectives:

- Learn how to install SQL Server 2016
- Learn how to install SQL Server Management Studio
- Learn how to create a database in SQL Server 2016
- 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 Microsoft SQL Server 2016 ODBC Client
- Learn how to import Microsoft Excel worksheet data into a database

What Is the Purpose of This Appendix?

In Chapter 1, we discussed the difference between personal and enterprise database management systems, and we introduced the personal DBMS Microsoft Access 2016 in depth in Chapter 1's section of "The Access Workbench."

In this appendix, we:

- Describe the enterprise class DBMS Microsoft SQL Server 2016.
- Describe how to install and use Microsoft SQL Server 2016 Developer Edition and the Microsoft SQL Server Management Studio.
- Describe and use DBMS data import techniques to import Microsoft Excel worksheet data into a Microsoft SQL Server 2016 database.

Note that you should work through this appendix in the following sequence:

- Before starting Chapter 3 on SQL, install Microsoft SQL Server 2016 and Microsoft SQL Server Management Studio by working through the appendix up to and including "How Do I Install the Microsoft SQL Server Management Studio?"

- As you work through the Chapter 3 sections on how to create and populate database tables, work up to and including “How Do I Use SQL Statements to Insert Database Data?” Both Chapter 3 and this appendix use the same WP database, and this work will show you how to create your own copy of the WP database.
- As you start to work through the Chapter 3 sections on how to use SQL Data Manipulation Language (DML) and SQL Data Definition Language (DDL), work through the section named “How Do I Work with SQL Queries in Microsoft SQL Server?” Both Chapter 3 and this appendix use the same WP database, and you can run the SQL Statements shown in Chapter 3 yourself, and see the results.
- As you work through the Appendix E material on SQL Views, you can run the SQL Statements shown in Appendix E yourself and see the results.
- As you work through the Appendix E material on SQL Persistent Stored Modules (SQL/PSM), you can run the SQL Statements shown in Appendix E yourself and see the results.
- Work through the section “How Do I Import Microsoft Excel Data into a Microsoft SQL Server Database Table?” in this appendix to understand how to import Microsoft Excel data in a database table.
- Work through the section “How Do I Create an ODBC Connection from Microsoft Access 2016 to an SQL Server 2016 Database?” in this appendix to understand how to use Microsoft Access as a development environment for a Microsoft SQL Server database.

What Is Microsoft SQL Server 2016?

Microsoft SQL Server is an enterprise-class DBMS that has been around for many years. SQL Server 2005, including **SQL Server 2005 Express Edition** was released in November 2005. **SQL Server 2008** and **SQL Server 2008 R2** (also with Express editions) were released in August 2008. **SQL Server 2012** was released in March 2012 and **SQL Server 2014** in March 2014. Now **SQL Server 2016** has been released and is available for use with this book.¹ Be aware that SQL Server 2016 is an enterprise-class DBMS and, as such, is much more complex than Microsoft Access. Further, it does not include application development tools, such as form and report generators.

The SQL Server Express Editions seem to be designed to compete with Oracle’s MySQL (see Appendix C). MySQL is an open-source database that, while not having as many features as SQL Server, has had the advantage of being available for free via download over the Internet. It has become widely used and very popular as a DBMS, supporting Web sites running the Apache Web server. SQL Server Express is becoming increasingly popular. Microsoft releases various versions of SQL Server, and while SQL Server Express is the least powerful version, it is intended for general use and can be downloaded for free. SQL Server 2016 Express is available in a basic package and in an advanced features package. The version

¹ As this book goes to press, the current version of SQL Server Express Edition is SQL Server 2016 Express. No service packs have been issued for SQL Server 2016. Always be sure you download and install the latest service pack for whichever version of SQL Server or any other DBMS you are using.

with advanced features is called **SQL Server 2016 Express Advanced**. Included in the advanced features package is support for SQL Server Reporting Services.

SQL Server 2016 is available in several versions, two of which—SQL Server 2016 Business Intelligence and SQL Server 2016 Web Developer—were introduced as part of SQL Server 2012. The basic set (SQL Server Web Developer is available only through Web site hosting companies such as Amazon Web Services [discussed in Chapter 8]) can be reviewed at the Microsoft SQL Server 2016 Web sites (www.microsoft.com/en-us/server-cloud/products/sql-server-editions/ and <https://www.microsoft.com/en-us/server-cloud/products/sql-server-2016/>) or in detail at the Microsoft Developer's Network Web site (<http://msdn.microsoft.com/en-us/library/ms143287.aspx>). For our purposes, there are five editions you need to be aware of:

- **Enterprise Edition.** This is a powerful and feature-laden commercial version. It handles up to the maximum number of CPUs or CPU cores allowed by the operating system, the maximum memory supported by the operating system, and a maximum database size of 524 Petabytes (PBytes). It includes full data warehouse capabilities and business intelligence capabilities.
- **Standard Edition.** This is the basic commercial version. It does not have the complete feature set of the Enterprise Edition. It handles up to 24 CPU cores, 128 GBytes of memory, and a maximum database size of 524 PBytes. It has some data warehouse and business intelligence capabilities.
- **Developer Edition.** This is a free, single-user version of the Enterprise Edition, and it has the complete feature set of the Enterprise Edition. It is intended, as the name implies, for use by a single user who is doing database and application development work.
- **Express Edition.** This free, feature-limited version is available for download. It supports 4 CPU cores, 1,410 MByte of memory, and a maximum database size of 10 GBytes. Despite its limitations, it is a great learning tool when the **Express Advanced** version, which includes the SQL Server 2016 reporting services package, is used.

On March 31, 2016, Microsoft announced that SQL Server Developer Edition would be available for free² instead of the approximately \$100 price it had previously had.

This is a significant announcement because it means that ***we can use Developer Edition instead of Express Edition as the foundation for our work in this book.*** Keep in mind, however, that you can use SQL Server Developer Edition only for your own, personal work. SQL Server Express Edition, on the other hand, can be used in a production environment.

² See <https://blogs.technet.microsoft.com/dataplatforminsider/2016/03/31/microsoft-sql-server-developer-edition-is-now-free/?MC=Vstudio&MC=SQL&MC=IE&MC=HTML5&MC=JavaScript>. This announcement initially applied to SQL Server 2014, the blog noted that it would also apply to SQL Server 2016.

Starting with SQL Server 2016, Microsoft has split the installation of the database engine and administrative tools into two separate installations. Previously, all needed components were installed in one installation procedure. And finally, Microsoft has announced that SQL Server will be available on the Linux operating system in the near future!³

By The Way

To install Microsoft SQL Server 2016, you will need to know whether you are running a 32-bit or 64-bit operating system. Windows 10—and earlier versions of the Windows operating system—is available in both 32-bit and 64-bit versions. To determine which version of Windows 10 you have, click the **Start** button (Windows icon on lower-left corner of the screen), then click **Settings** (the gear icon), then click **System**, and then click **About**.

The 32-bit versions of Microsoft programs (including, but not limited to, SQL Server 2016 Express) at the download sites are designated as **x86**, while the 64-bit versions are designated as **x64**. The “x86” refers to Intel processors that include “86” in the processor name (for example, the Intel 80486 CPU chip) and related processors such as the Intel Pentium CPU chip (which would have been the 80586 if Intel hadn’t switched to names instead of numbers for its product line).

In this appendix, we are using a 64-bit version of Windows 10 Anniversary Update (Version 1607), updated with current security and operating system updates. All instructions are based on this version of Windows 10, and may vary if you are using a different Windows edition (7 or 8.10) or an earlier version of Windows 10.

Why Should I Learn to Use SQL Server?

For the purposes of this book, the most important reason to learn to use Microsoft SQL Server 2016 is that SQL Server really handles SQL well. All the SQL results shown in Chapter 3 and Appendix E were created in SQL Server 2016. 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 SQL Server.

What Will This Appendix Teach Me?

As its title implies, this appendix is designed to get you started creating databases and running SQL commands so that you can use a more robust SQL environment than that provided by Microsoft Access.

³ See <https://www.microsoft.com/en-us/server-cloud/sql-server-on-linux.aspx>. Given Microsoft’s history of having Microsoft products run only on the Microsoft Windows operating system (OS), this is a very important announcement.

What Won't This Appendix Teach Me?

The material in this appendix does not go beyond what is necessary to get you started. There are many important SQL Server topics not covered here, including stored procedures, triggers, backups and restores, and database security. These topics are covered in David M. Kroenke and David J. Auer, *Database Processing: Fundamentals, Design, and Implementation*, 14th edition (Upper Saddle River, NJ: Prentice Hall, 2016).

How Do I Install SQL Server 2016 Developer Edition?

To install SQL Server 2016 Developer edition on Microsoft Windows 10, you may need to download the following programs:

1. **.Net Framework 3.5 Service Pack 1.** This is provided with the Windows 10 OS and is an update to Microsoft .NET Framework 3.5. This version of the .NET Framework is *not* automatically installed. If you are running Windows 10, this can be installed by using the Windows Control Pane. Open Control Panel using the **Windows key + x key combination** to display the shortcut menu, then click the **Control Panel** command. In Control Panel, click **Programs**, and then click **Turn Windows features on or off**. In the Windows Features dialog box, select the .NET Framework 3.5 features (and additional .NET Framework 4.6 Advanced Services) shown in Figure A-1. Click the **OK** button to begin the installation process.
2. **.Net Framework 4.6.1.** This is an updated version of the Microsoft .NET Framework (<http://www.visualstudio.com/downloads/download-visual-studio-vs>). If you are running Windows 10, this is provided with the Windows OS and should be enabled by default. If needed, in Windows 10, this can be installed by using the Windows Control Pane. Open Control Panel using the **Windows key + x key combination** to display the shortcut menu, then click the **Control Panel** command. In Control Panel, click **Programs**, and then click **Turn Windows features on or off**. In the Windows Features dialog box, select the .NET Framework 4.6 features. Click the **OK** button to begin the installation process.

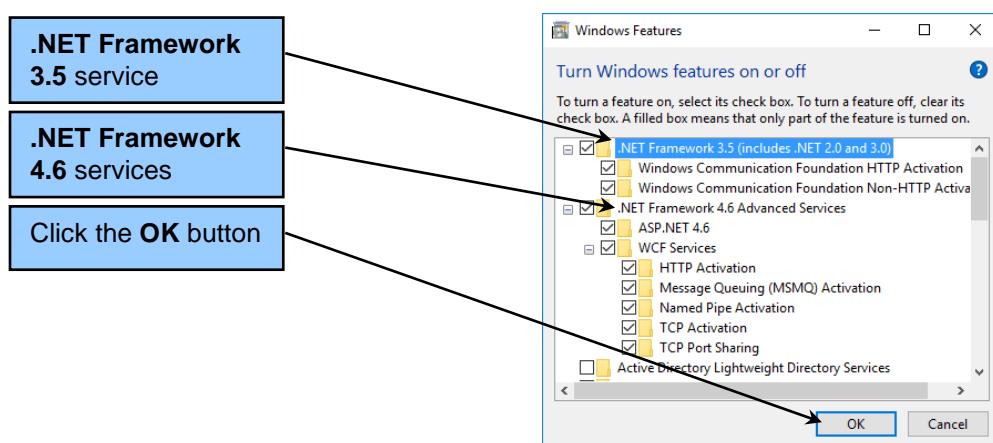


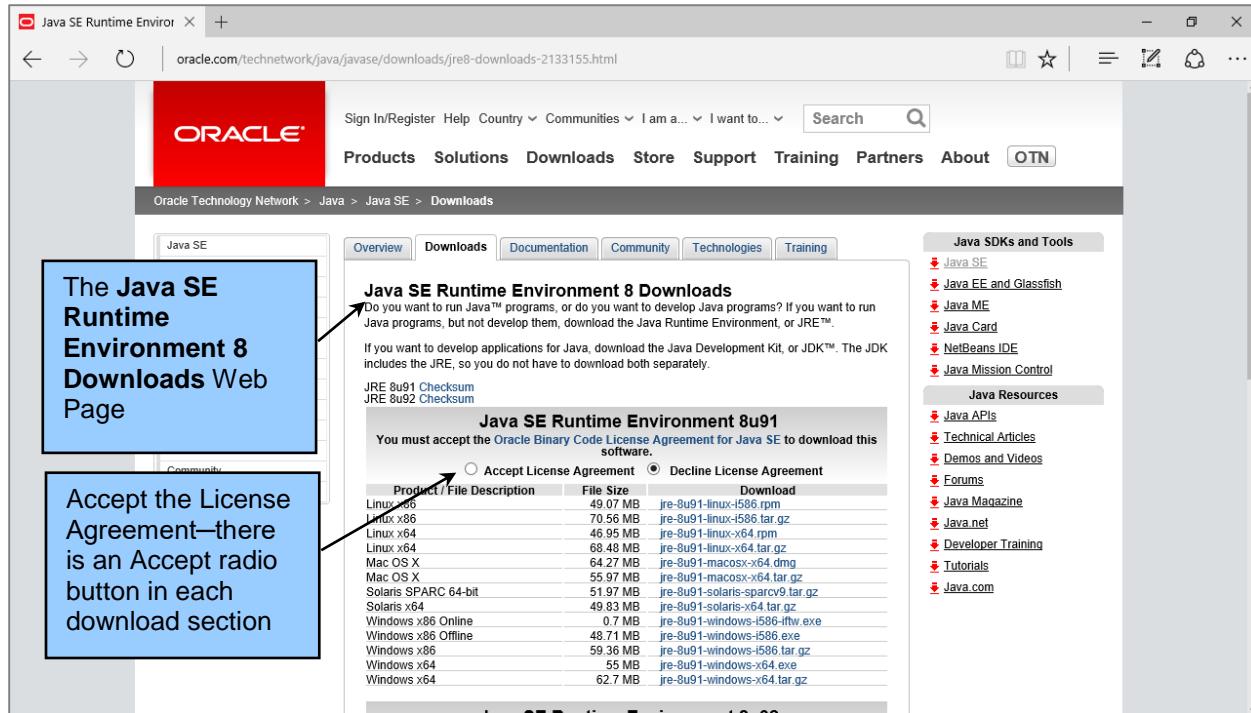
Figure A-1 — The Windows Features Dialog Box

1. **Oracle Java Runtime Environment (JRE) 7 Update 51 or higher.** This is required in order to install the PolyBase component of Microsoft SQL Server 2016. The current version of the JRE (which is JRE 8 update 111 as of January 10, 2017) can be found at <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>. Download the appropriate JRE file into your Downloads folder as shown in Figure A-2. Note that you may have to scroll down on the Web page to find the latest JRE version, as shown in Figure A-2(b). When the download is finished, click the **Run** button shown in Figure A-2(b), and install the JRE.
2. **Update for Visual C++ 2013, Visual C++ Redistributable Package, and KB3164398.** This is a required patched version of the Visual C++ 2013 redistributable package. It can be found at <https://support.microsoft.com/en-us/kb/3138367>. Download both the 32-bit (x86) and 64-bit versions (x64) into your Downloads folder, and then install (right click the filename as you will need to use the *Run as Administrator* command). Figure A-3 shows the 32-bit version of the update being installed. As of January 10, 2017, Microsoft Visual C++ 2013 Update 5 is available. Reboot your computer after installing (you may be prompted to do so).
3. **Microsoft SQL Server 2016 Developer Edition.** The **Microsoft SQL Server 2016 Management Studio** and other utilities are included, or these utilities are linked to, in the download of Microsoft SQL Server 2016 Developer Edition but require a separate installation. Microsoft SQL Server 2016 Management Studio is the graphical management utility for Microsoft SQL Server 2016. Microsoft SQL Server 2016 is a command line-oriented program. Microsoft SQL Server 2016 Management Studio makes it much easier to work with SQL Server.
4. **Critical update for SQL Server 2016 MSVCRT prerequisites.** Download from <https://support.microsoft.com/en-us/kb/3164398>. This is a critical update, and patches the Visual C++ package as well as SQL Server 2016.

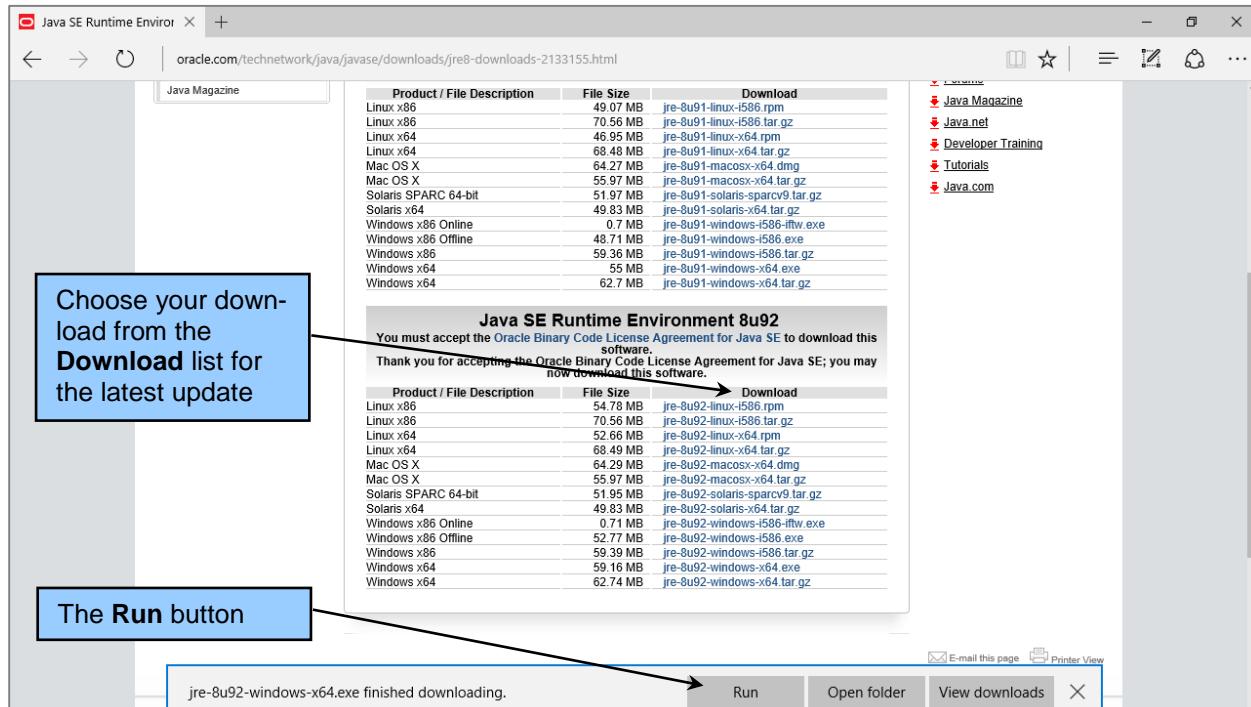
To download SQL Server 2016 Developer Edition:

Note: Microsoft recently changed the download steps, and although the steps shown here are current be aware that they may change again!

1. **Join the Microsoft Visual Studio Dev Essentials Program.** This program is free. Go to <https://www.visualstudio.com/free-developer-offers/> and click the **Join now** button. Complete the registration steps as requested. When you are done, the Welcome to your Visual Studio Dev Essentials Web page will be displayed, as shown in Figure A-4—note that the tab reads *Home-My Visual Studio*.
2. **Download Microsoft SQL Server 2016 Developer Edition.** In the Microsoft SQL Server Developer Edition box, click the **Download** button. A horizontal dialog box appears on the Home – My Visual Studio Web page asking if you want to save the **SQLServer2016-SSEI-Dev.exe** file. Click the **Save** button.
3. The basic installation program is downloaded into the Windows 10 **This PC | Downloads** folder. A security scan is run on the file, and the *{filename} finished downloading* message shown in Figure A-6 is displayed. Click the **Close** button. Close the **Visual Studio Dev Essentials Web Page**.



(a) The Java SE Runtime Environment 8 Downloads Web Page



(b) The Run Button

Figure A-2 — Installing the Java Runtime Environment (JRE)

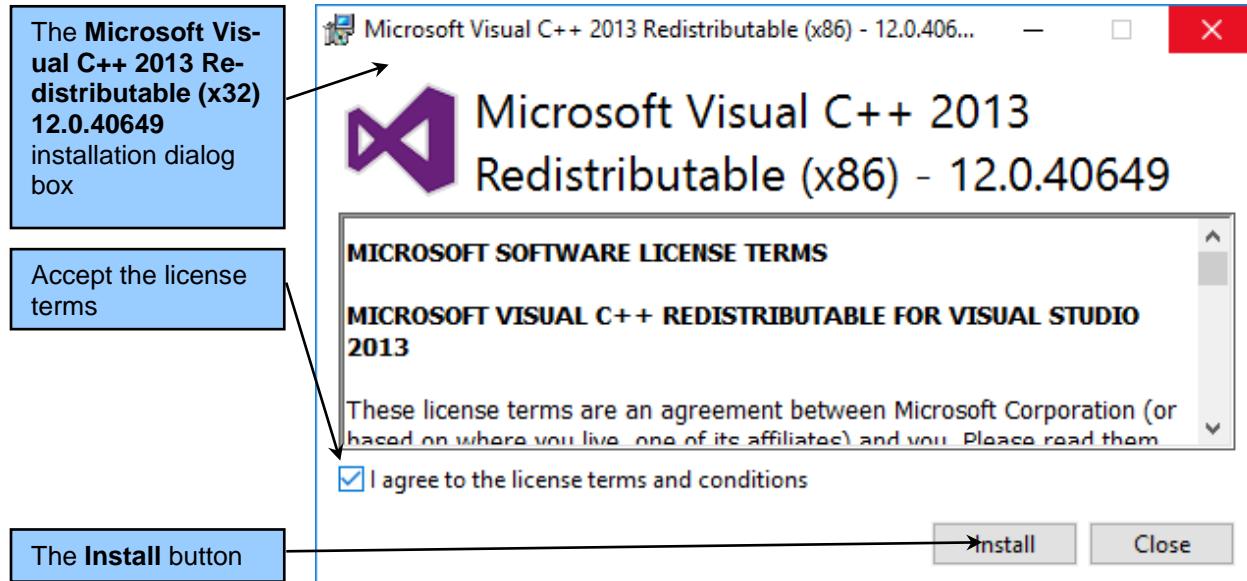


Figure A-3 — The Microsoft Visual C++ 2013 Redistributable (x32) Installation

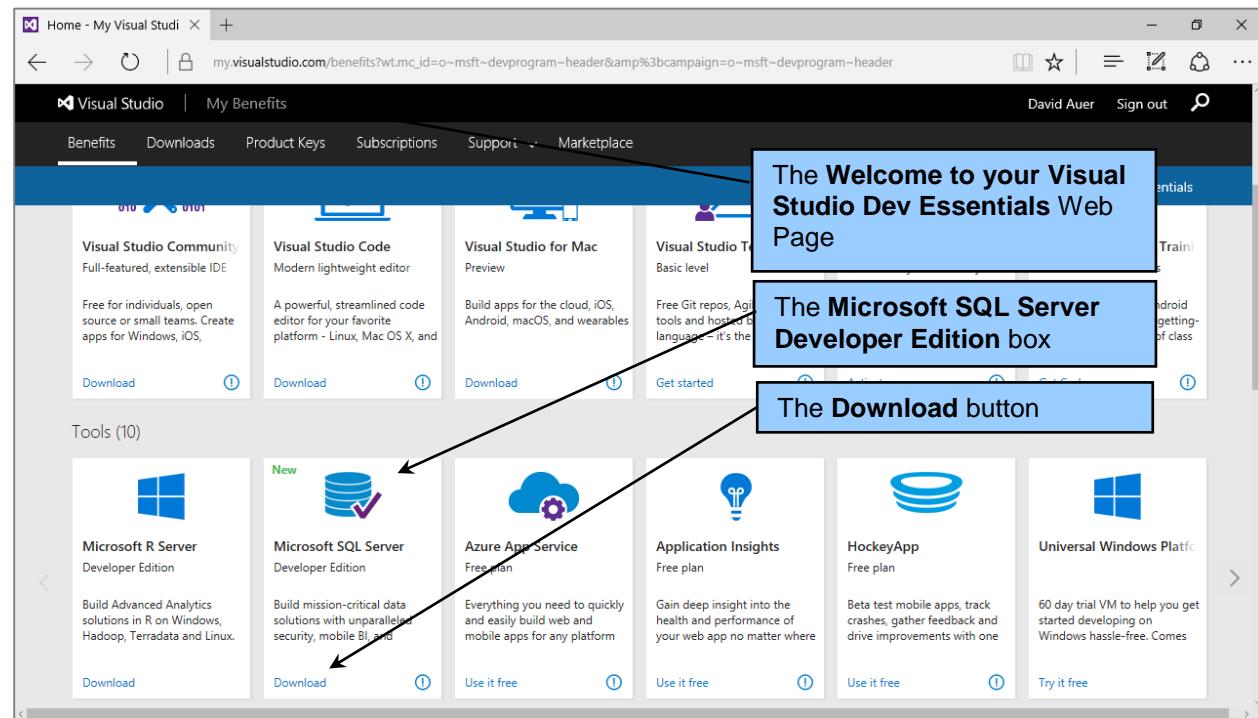


Figure A-4— The Welcome to Your Visual Studio Dev Essentials Web Page

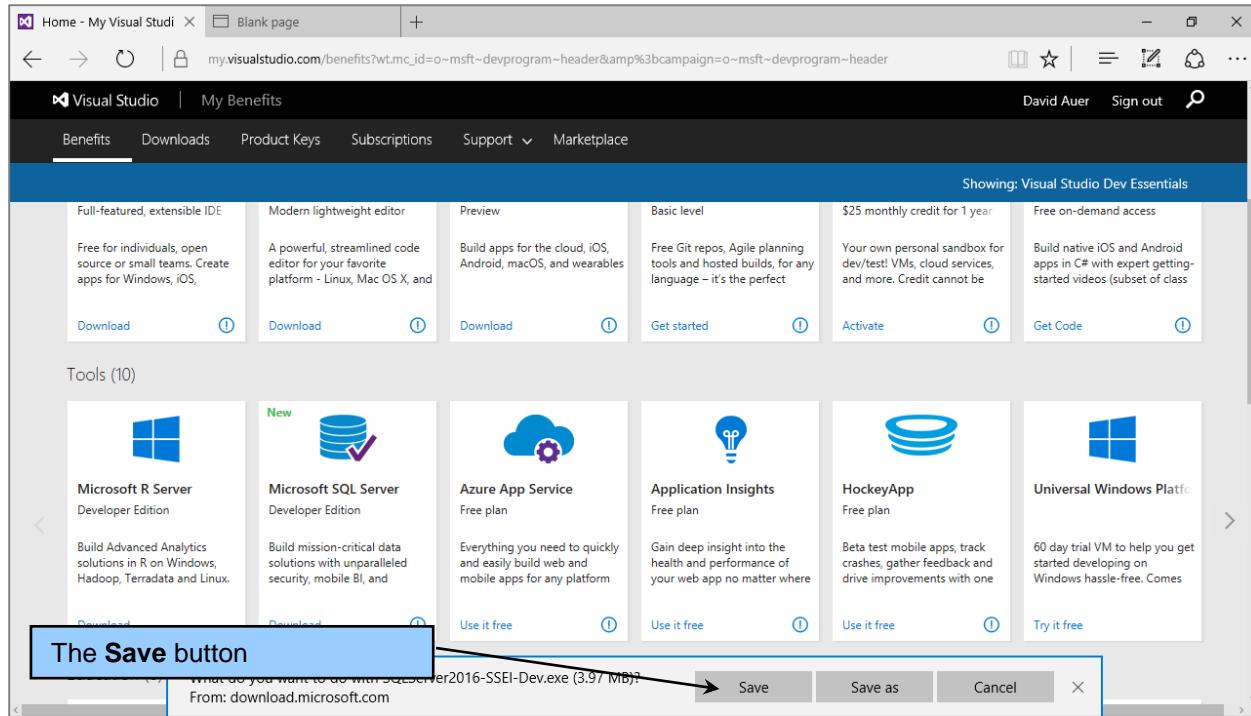


Figure A-5 — The Welcome to Visual Studio Dev Essentials! Web Page

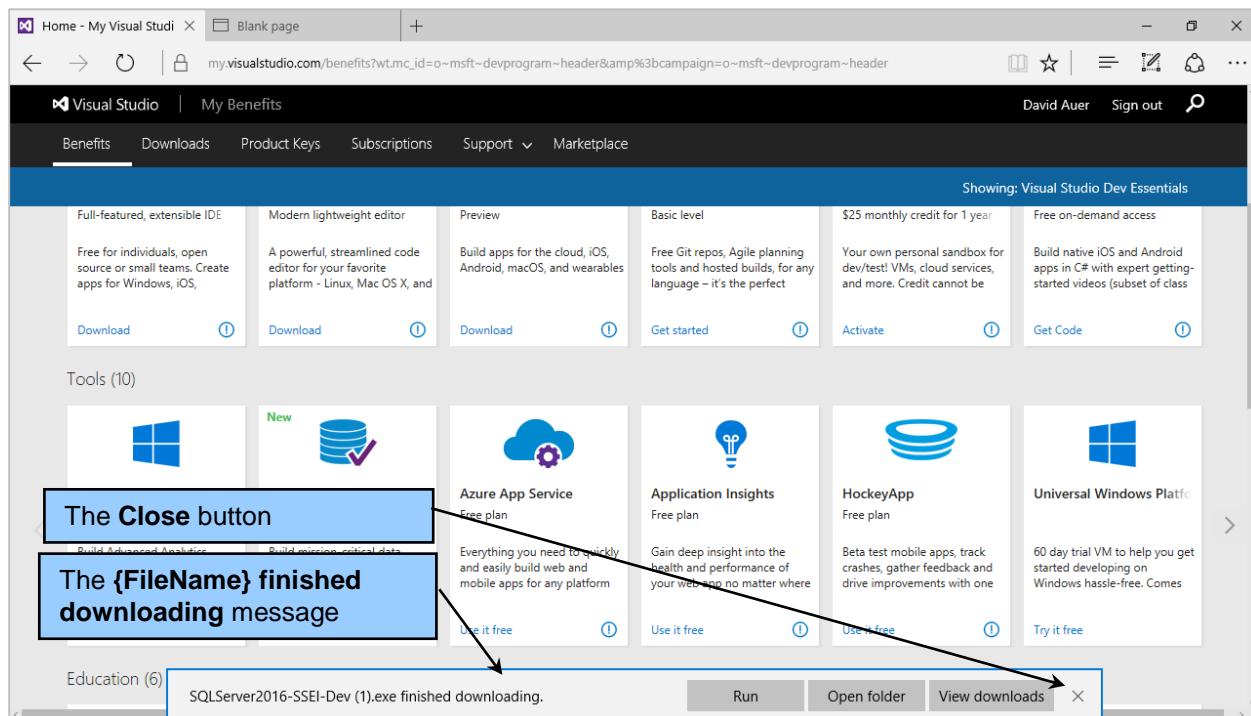


Figure A-6 — The {FileName} Finished Downloading Messaged

Starting the SQL Server 2016 Installation:

1. The file we need to use to **download the actual SQL Server 2016 installation file** is now in our **This PC | Downloads** folder. Open **Microsoft File Explorer**, and browse to the **This PC | Downloads** folder. Locate the **SQLServer2016-SSEI-Dev.exe** file. Right-click the file name to display the shortcut menu, and then click the Run as administrator command as shown in Figure A-7(a).
2. The **SQL Server 2016 with SP1 Developer Edition** dialog box **Select an installation type** page is displayed, as shown in Figure A-7(b). Click the **Download Media** button.
3. The SQL Server 2016 with SP1 Developer Edition dialog box **Specify SQL Server installer download** page is displayed, as shown in Figure A-7(c). The settings are correct, because we want to download the English language ISO file into the Downloads folder. Click the **Download** button.
4. The SQL Server 2016 with SP1 Developer Edition dialog box **Download successful!** page is displayed, as shown in Figure A-7(d). The file we need has been downloaded into the Downloads folder. Click the **Open folder** button.
5. There is now a file named **SQLServer2016SP1-FullSlipstream-x64-ENU-DEV.iso** in the Downloads folder, as shown in Figure A-7(e). The file extension ***.iso** indicates that is file is a package of folders and files and is generally intended for burning a CD or DVD that will then be used for the actual program installation. However, Windows 10 allows us to directly use an ***.iso** file by mounting the file as we would a CD or DVD. Right-click the file name to display the shortcut menu and the click the **Mount** command.
6. This opens a virtual DVD drive, with all folders and files available, as shown in Figure A-7(f). Right-click the **setup.exe** file, and click the **Run as administrator** command. When the User Account Control dialog box displayed, click the **Yes** button.

Note that the actual version of SQL Server 2016 Developer edition we have downloaded includes Server Pack 1 (SP1). The exact version of SQL server you download and install will depend upon what patches Microsoft has integrated in the version of SQL Server 2016 that is posted to their download Web site.

Figure A-8 illustrates the installation process of Microsoft SQL Server on a Windows operating system. Figure A-8(a) shows the **SQL Server Installation Center** dialog box, which is the first screen to appear when the SQL Server 2016 Developer installation process is begun after we the Yes button in step 6 above. After the SQL Server Installation Center dialog box is displayed, we clicked on the **Installation** button to display the *Installation* screen as shown in Figure A-8(a).

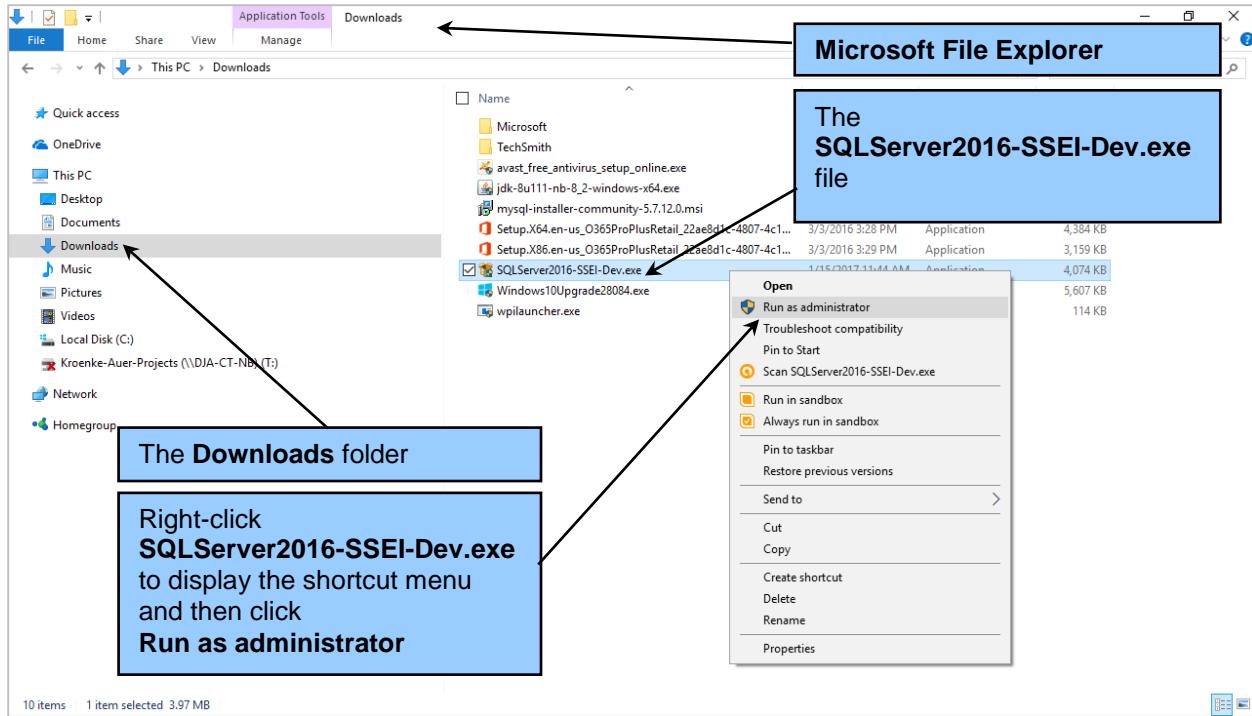


Figure A-7(a) — Running the SQLServer2016-SSEI-Dev.exe Program

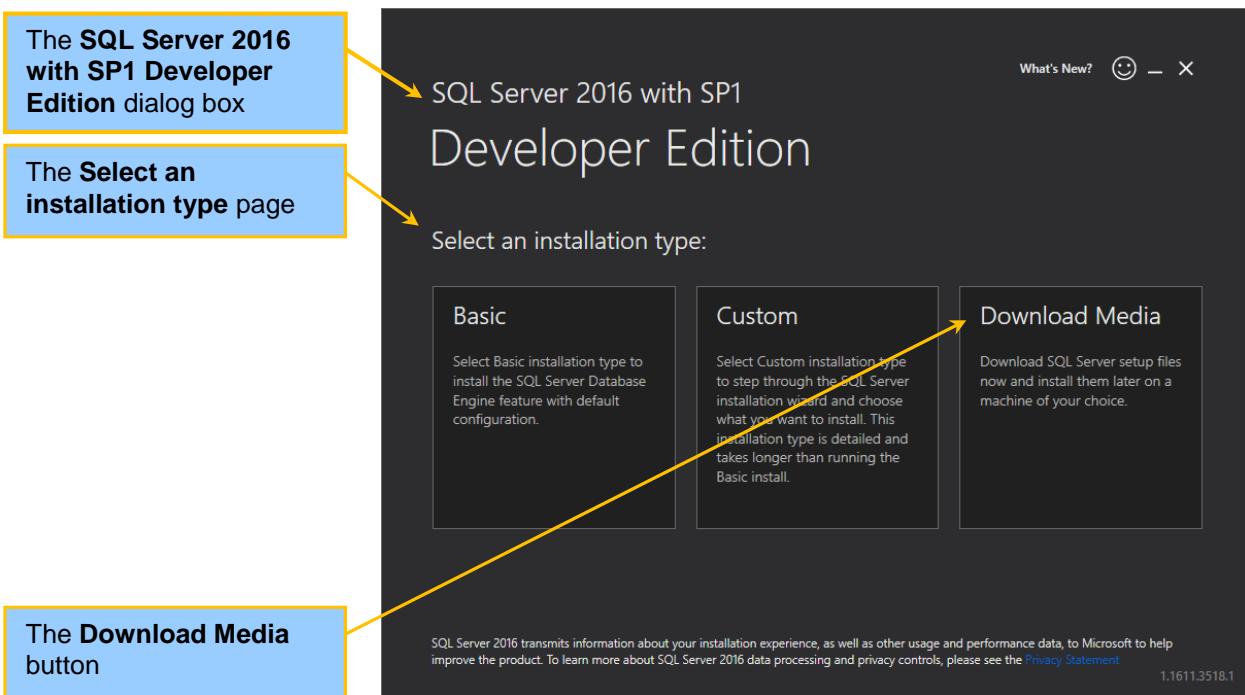


Figure A-7(b) — The SQL Server 2016 with SP1 Developer Edition Dialog Box

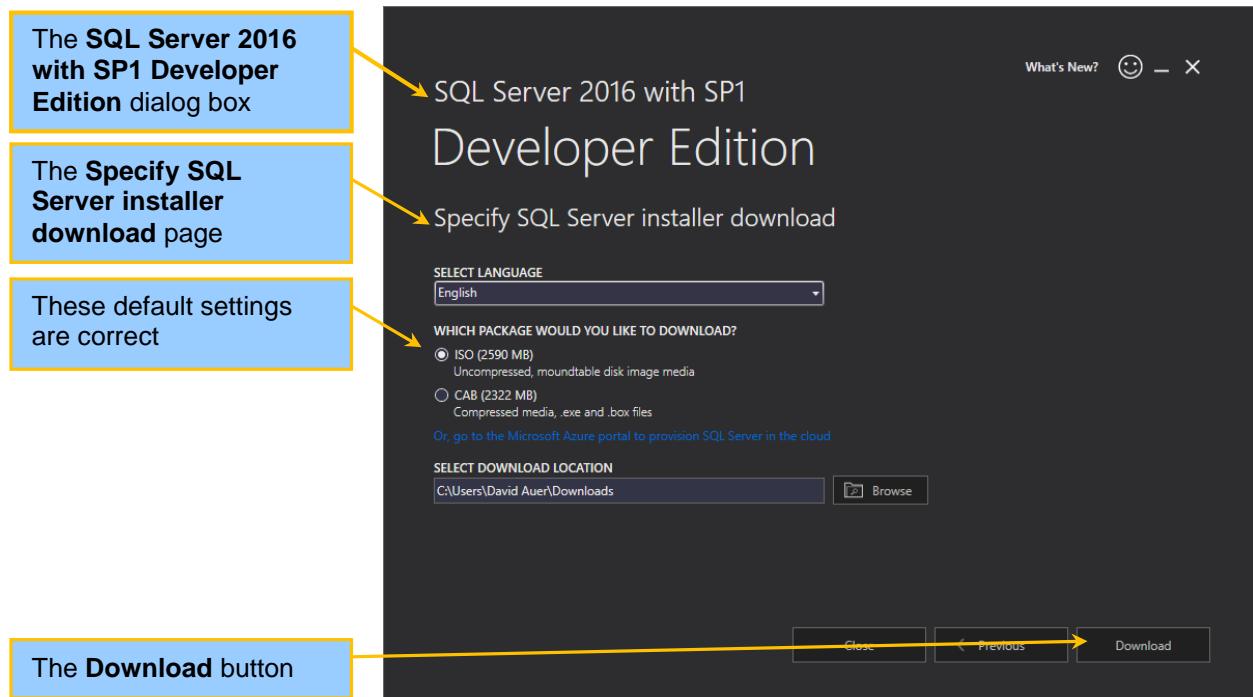


Figure A-7(c) — The SQL Server Installer Download Page

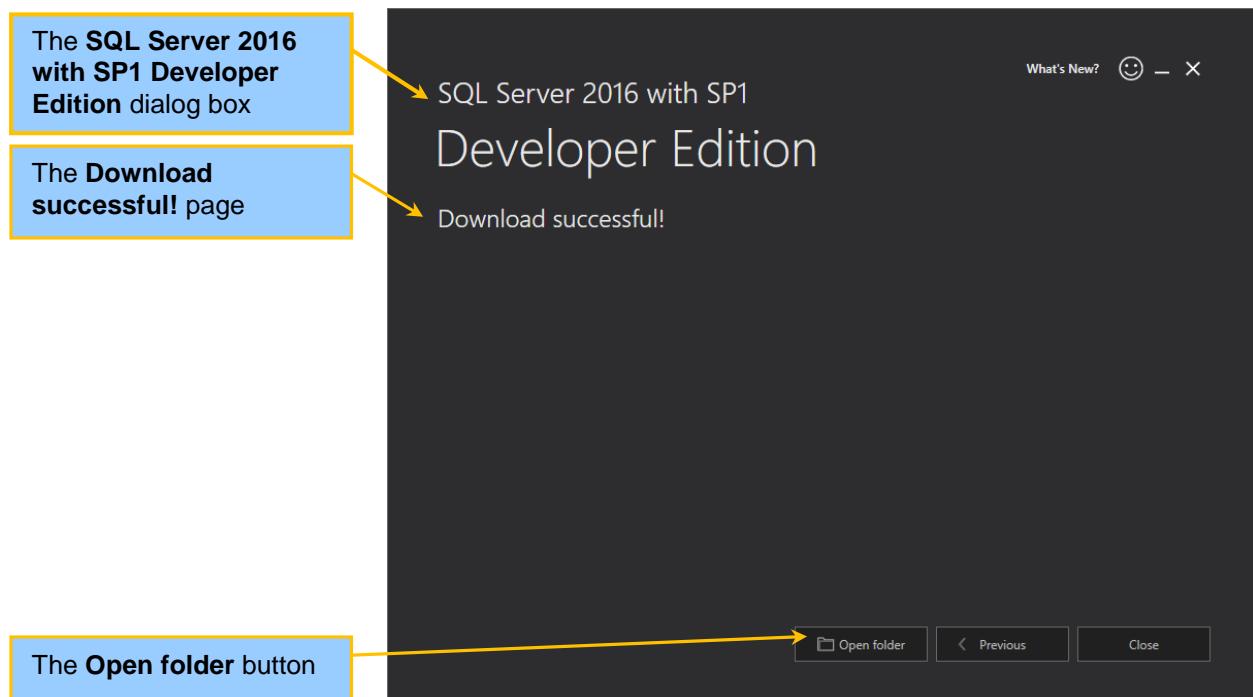


Figure A-7(d) — The Download Successful! Page

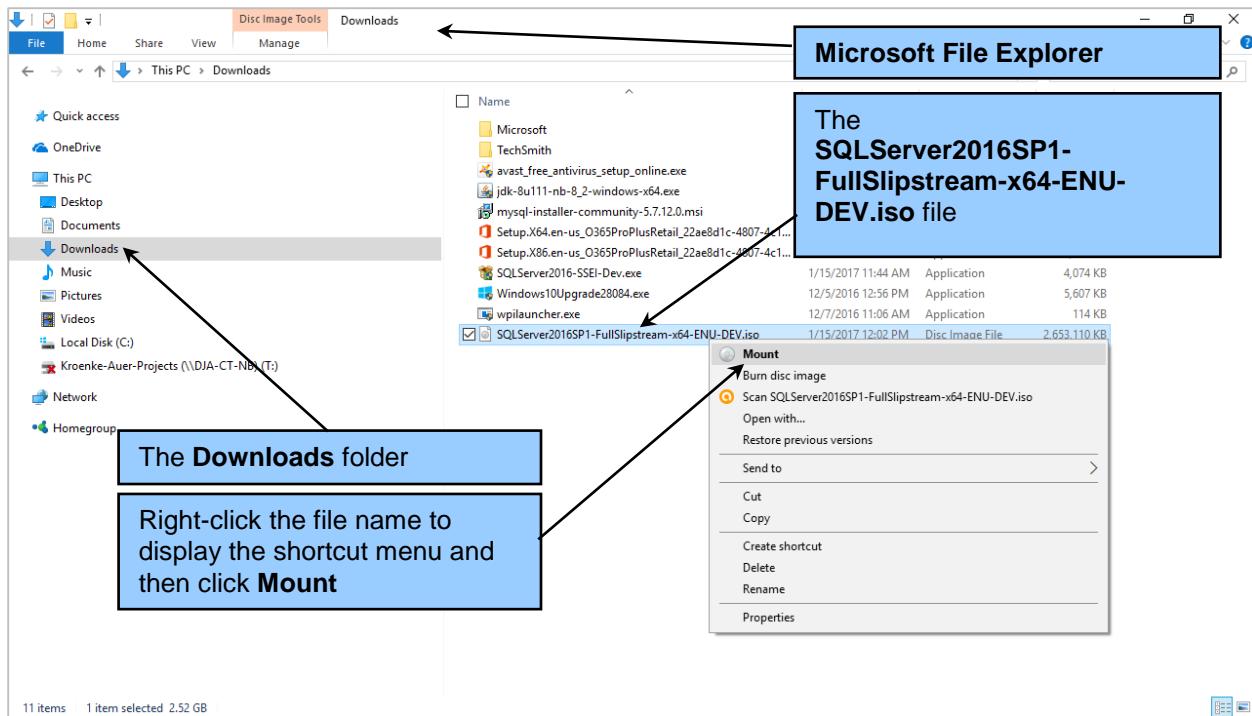


Figure A-7(e) — Mounting the SQLServer 2016 ISO file

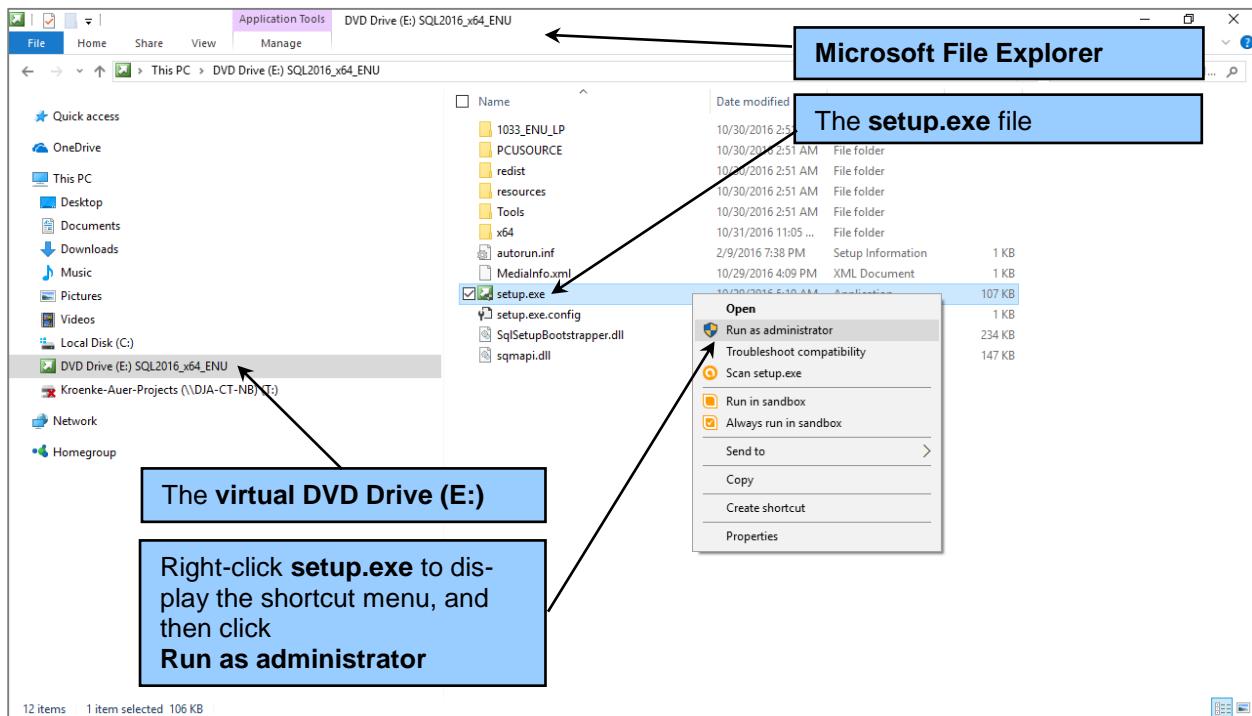


Figure A-7(f) — The Virtual DVD Drive and the Setup.exe File

Installing SQL Server 2016:

1. To start the actual installation process, click on **Installation** button in the left hand page if necessary to move to the Installation page, and then click **New SQL Server stand-alone installation or add features to an existing installation** button in the SQL Server Installation Center, as shown in Figure A-8(a). The SQL Server 2016 Setup window appears with the Product Key page displayed, as shown in Figure A-8(b).
2. Select the edition of SQL Server 2016 that you are installing, which for us is **Developer** edition. *SQL Server 2016 Developer Edition*, which provides full functionality for a single person in a non-production environment, maybe installed as a free edition. *SQL Server 2016 Express edition* is also a free edition, does not need a product key, and can be used in multiuser, production environment. If you are installing the Enterprise version that requires a product key, type it in the *Enter the product key* text box. If you install Enterprise without a product key, you will install a 180-day trial version.
3. Click the **Next** button. The SQL Server 2016 Setup **License Terms** page is displayed, as shown in Figure A-8(c). Accept the license terms.
4. Click the **Next** button. The installation works through the process. The SQL Server 2016 Setup **Global Rules** page is shown in Figure A-8(d). Note that if all the rules are passed, the installation procedure will skip displaying this screen! If you want to see it, hit the *Back* button and then the *View Details* button.

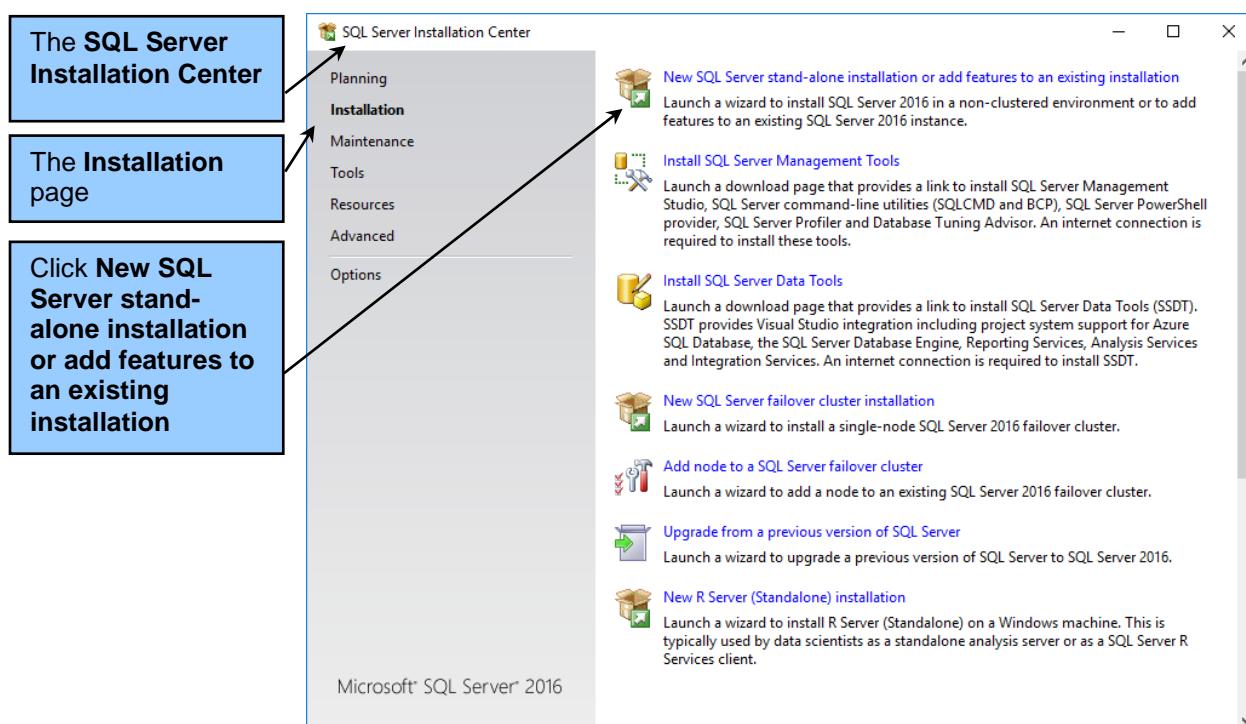


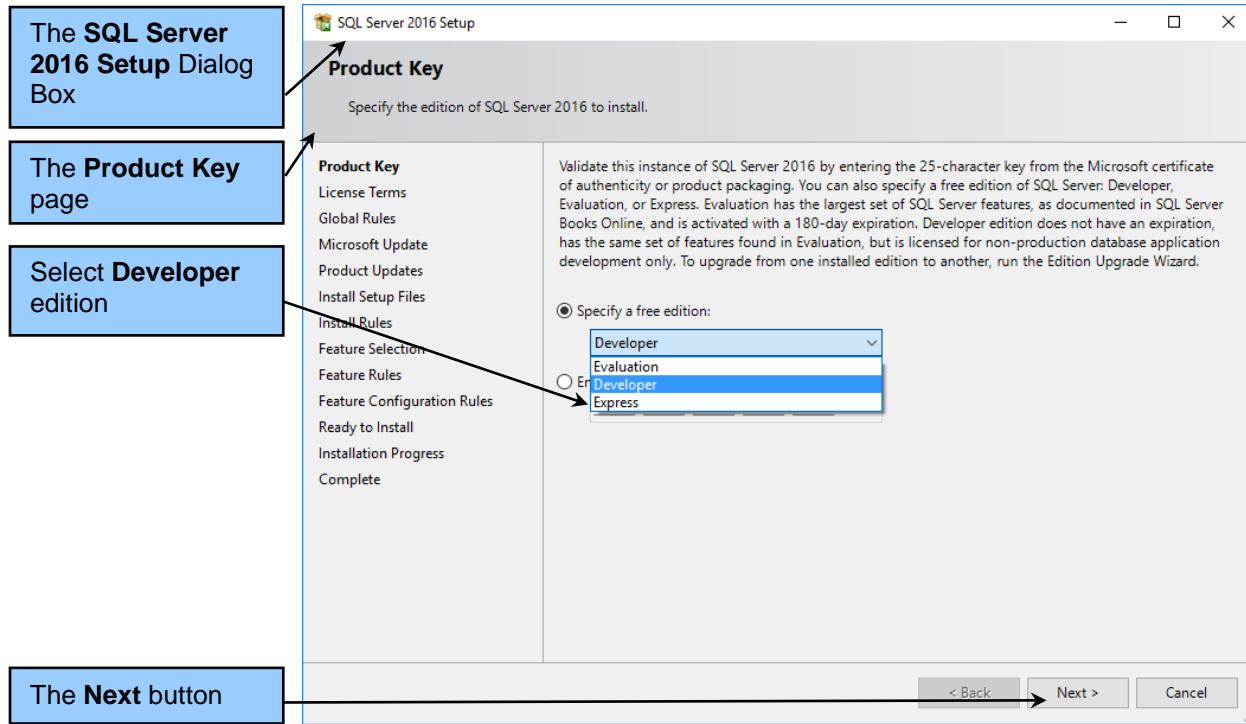
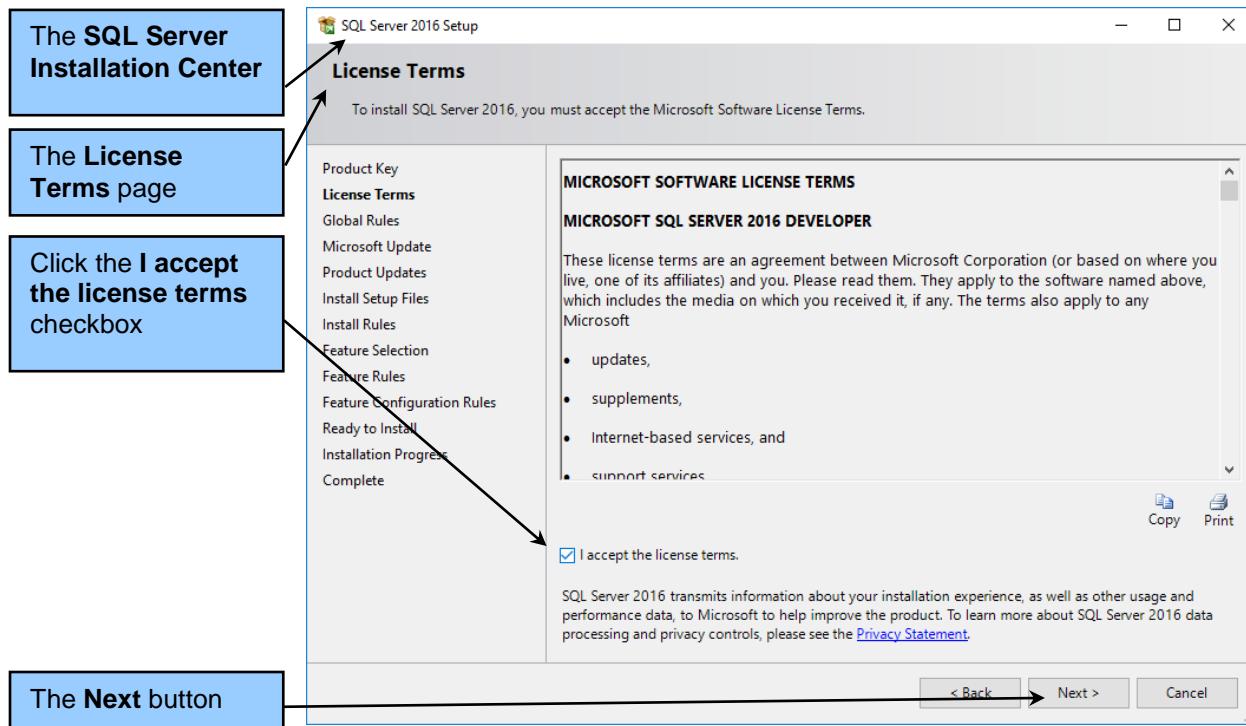
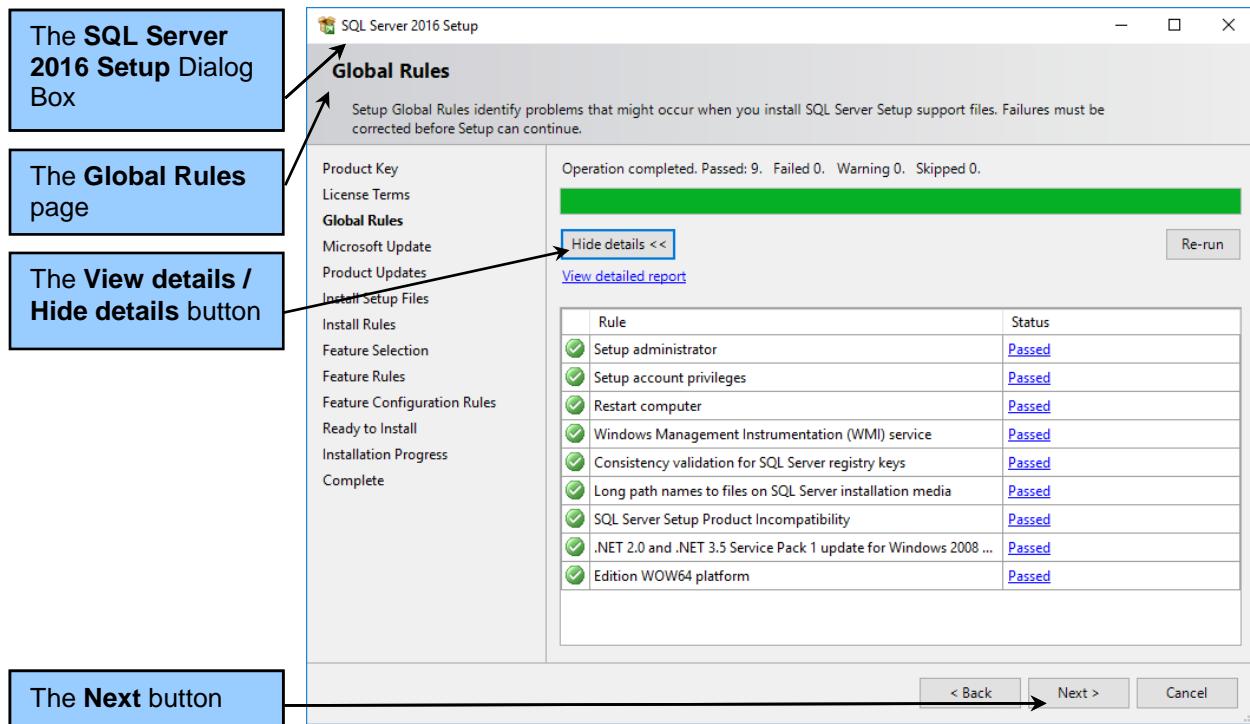
Figure A-8(a) — The SQL Server Installation Center Dialog Box Installation Page**Figure A-8(b) — The SQL Server Installation Center Dialog Box Installation Page**

Figure A-8(c) — The SQL Server Installation Center Dialog Box License Terms Page**Figure A-8(d) — The SQL Server Installation Center Dialog Box Global Rules Page**

5. The SQL Server 2016 Setup **Microsoft Update** page is displayed, as shown in Figure A-8(e). Check the *Use Microsoft Update to check for updates* checkbox.
6. Click the **Next** button. If there are product updates available, the SQL Server 2016 Setup **Product Updates** page is displayed, as shown in Figure A-8(f). This page shows the updates that will be installed as part of the installation. *If there are no updates available, this page will not be displayed.*
7. If the **Product Updates** page is displayed, click the **Next** button. The installation then installs needed setup files, as shown on the SQL Server 2016 Setup **Install Setup Files** page, as shown in Figure A-8(g). *This page is displayed only during the file installation process.* The **Back** button is not available to go back to this page!
8. The installation works through the install process until the SQL Server 2016 Setup **Install Rules** page is displayed, as shown in Figure A-8(h). *This page is displayed only if there are rules that did not pass.* If this page is not displayed, press the **Back** button, and then the **View Details** button. All rules should pass their tests. The Windows Firewall warning simply says that the Windows Firewall is enabled on the computer and is a reminder to set the correct port settings in the firewall depending upon what access you want to allow. The default settings are enough for now.
9. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Feature Selection** page is displayed. Click the **Select All** button to install all features, as shown in Figure A-8(i). Note that this selection resets the installation outline in the left hand window, and more steps are displayed!

10. Click the **Next** button. The installation works through the SQL Server 2016 Setup **Feature Rules** page, as shown in Figure A-8(j). *Note that if there are no problems, this page will not be displayed.* If you installed all the .NET Framework products, the Java JRE, and the Visual Studio C++ 2013 Redistributable files specified above, the installation will pass these tests as shown. All the setting and selections are correct here, so we will leave everything as selected.

- If you have any failed tests in feature rules, cancel the SQL Server installation, install all the software prerequisites listed above, and restart the installation.

11. If you are on the **Feature Rules** page, click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Instance Configuration** page is displayed, as shown in Figure A-8(k). If you are installing the Enterprise, Standard, or Developer version of SQL Server 2016, the Default instance radio button will be selected and the instance name of *MSSQLSERVER* will be used. If you are installing an SQL Server 2016 Express Edition, the Named instance radio button will be selected and the instance name of *SQLEXPRESS* will be used. The term **instance** refers to the fact that more than one copy of SQL Server 2016 can be installed on one computer, and each separate installation is referred to as an instance. For example, it is possible (and quite easy) to install an instance of the developer version of SQL Server 2016 (functionally equivalent to the enterprise version but intended for a single user) and an instance of SQL Server 2016 Express edition on the same workstation. The settings that are displayed will be correct for your installation, so there is nothing to change here.

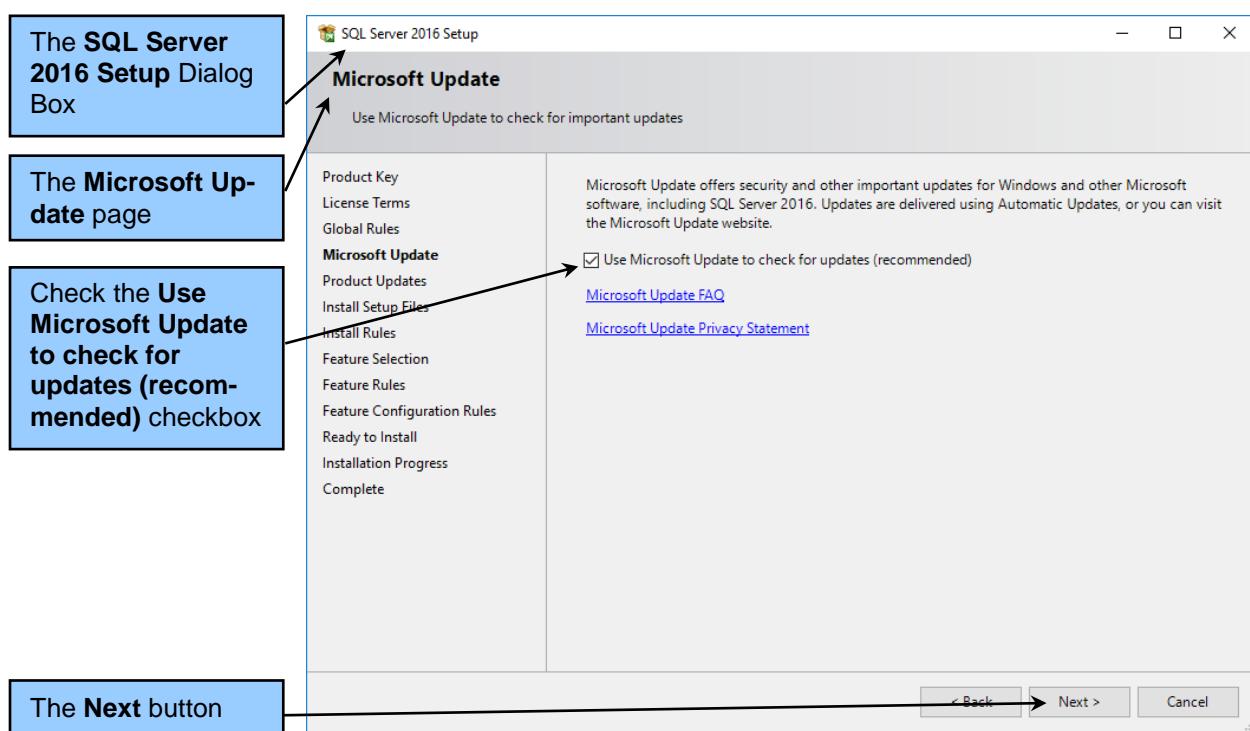


Figure A-8(e) — The SQL Server Installation Center Dialog Box Microsoft Update Page

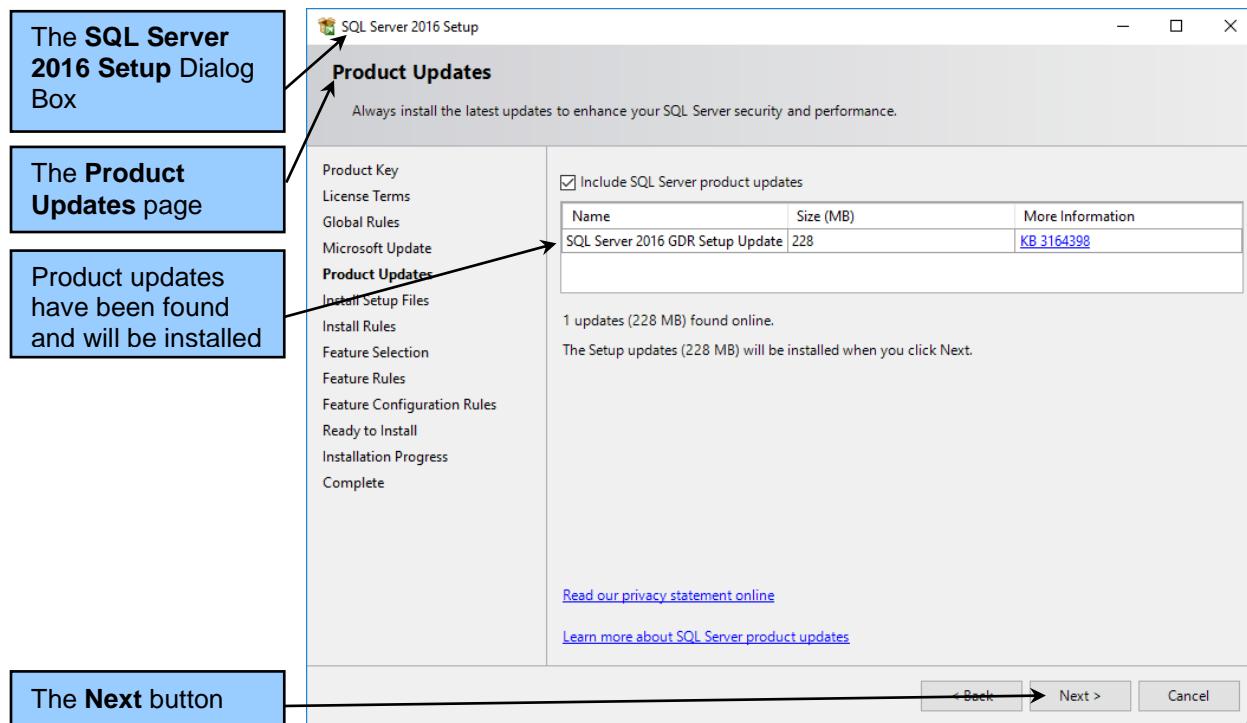


Figure A-8(f) — The SQL Server Installation Center Dialog Box Product Updates Page

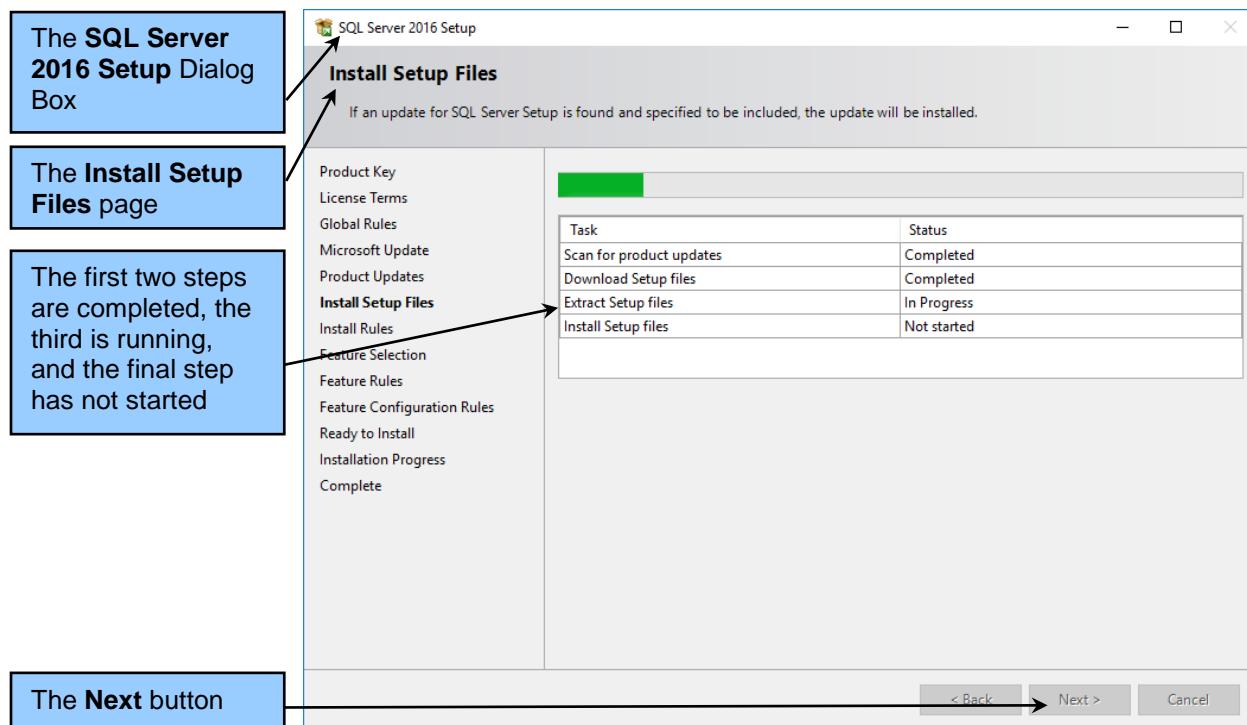


Figure A-8(g) — The SQL Server Installation Center Dialog Box Install Setup Files Page

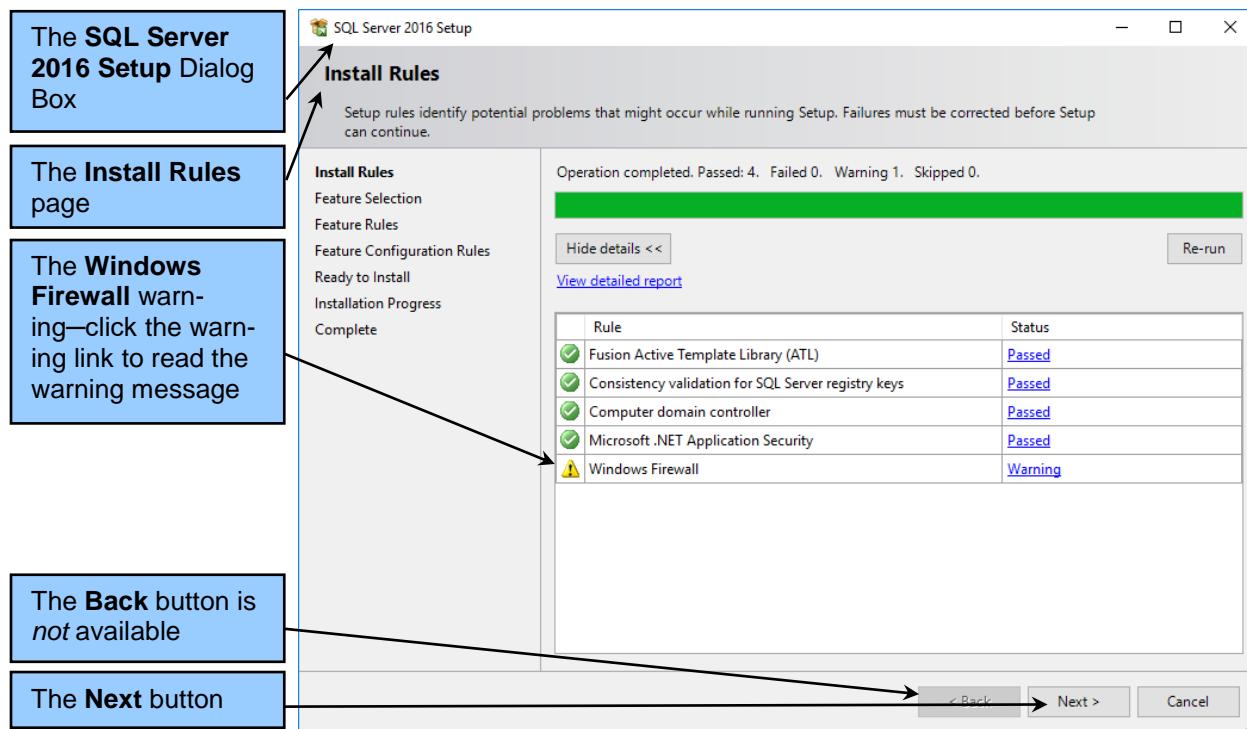


Figure A-8(h) — The SQL Server Installation Center Dialog Box Install Rules Page

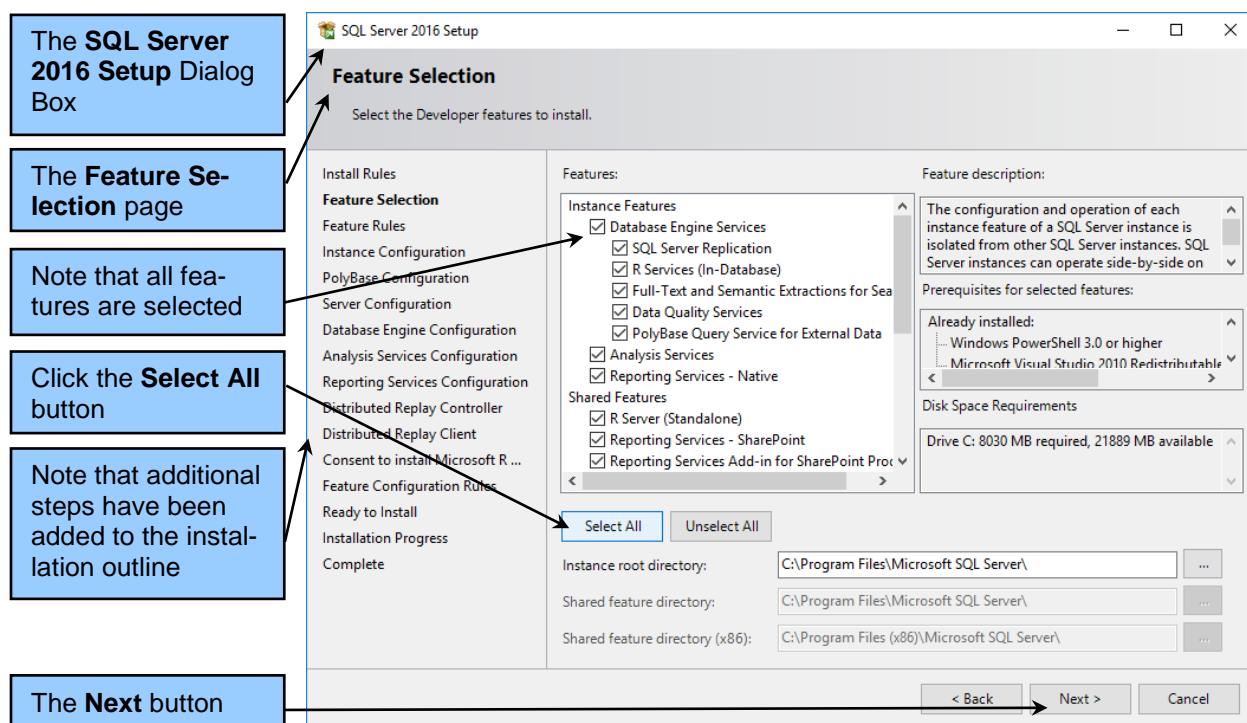


Figure A-8(i) — The SQL Server Installation Center Dialog Box Installation Page

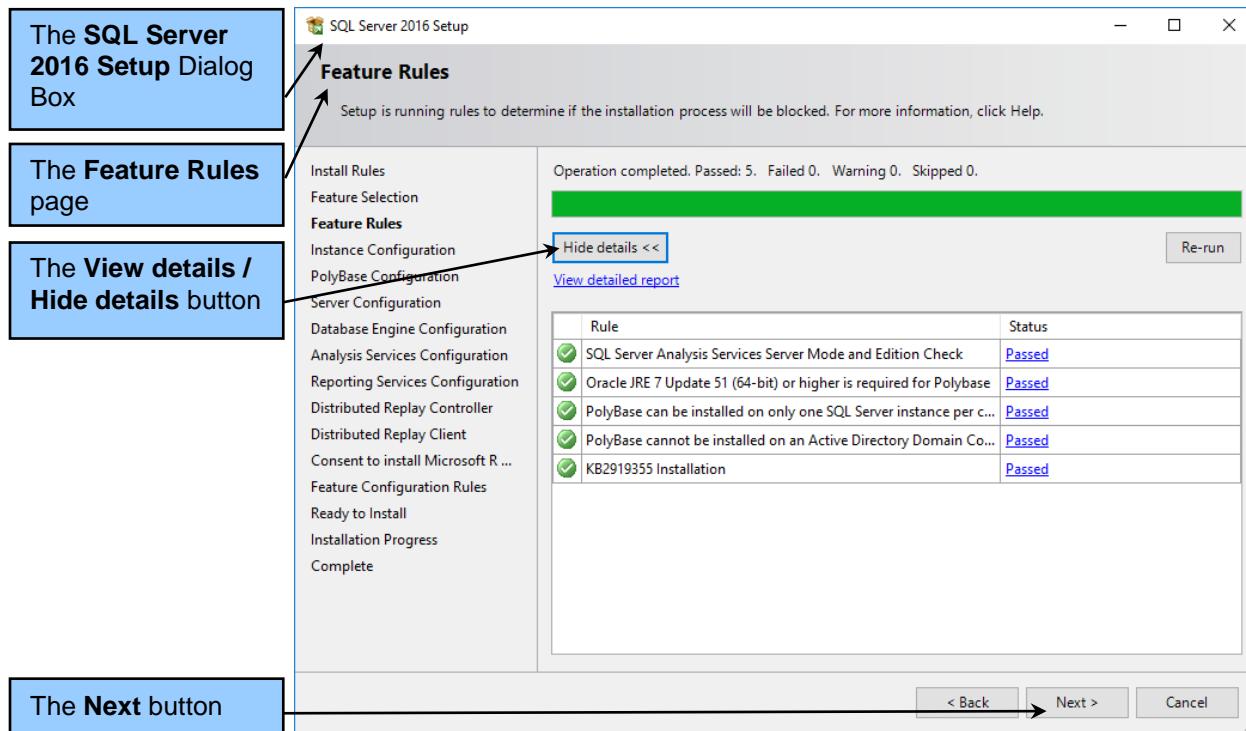


Figure A-8(j) — The SQL Server Installation Center Dialog Box Feature Rules Page

12. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **PolyBase Configuration** page is displayed, as shown in Figure A-8(l). PolyBase is a Microsoft technology that is used in Microsoft's Azure cloud computing environment (see <http://www.jamesserra.com/archive/2014/02/polybase-explained/>). The default settings are correct for our installation.
13. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Server Configuration** page is displayed, as shown in Figure A-8(m). Change (1) the Startup Type for **SQL Server Agent** from Manual to **Automatic**, and (2) the **SQL Server Browser** from Disabled to **Automatic**, as shown in Figure A-8(m).
14. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Database Engine Configuration** page is displayed, as shown in Figure A-8(n). Change the **Authentication Mode** to **Mixed Mode**, and enter and confirm a **password** for the SQL Server system administrator (sa) [**be sure you remember this password!**]. Click the **Add Current User** button to add *yourself* as an SQL Server administrator. All the other settings are correct.
15. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Analysis Services Configuration** page is displayed, as shown in Figure A-8(o). Keep the selected server mode, and click the **Add Current User** button to give yourself administrative permissions for Analysis Services. All the other settings are correct.

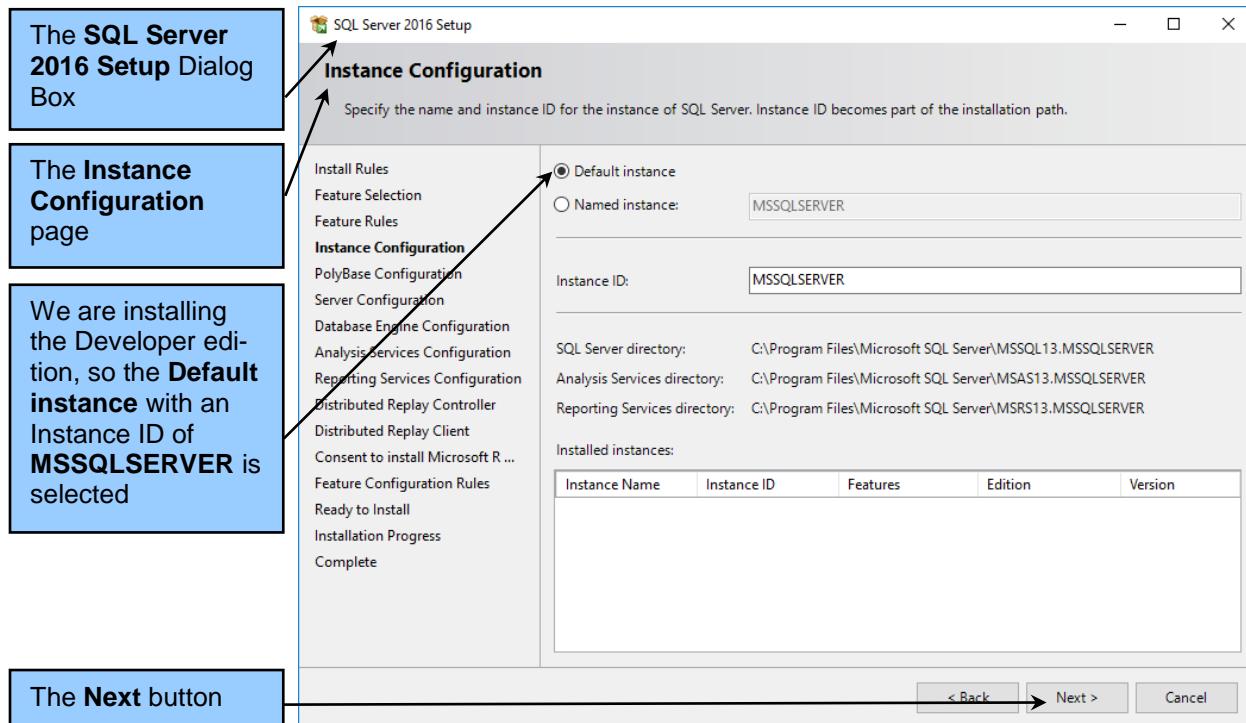


Figure A-8(k) — The SQL Server Installation Center Dialog Box Feature Rules Page

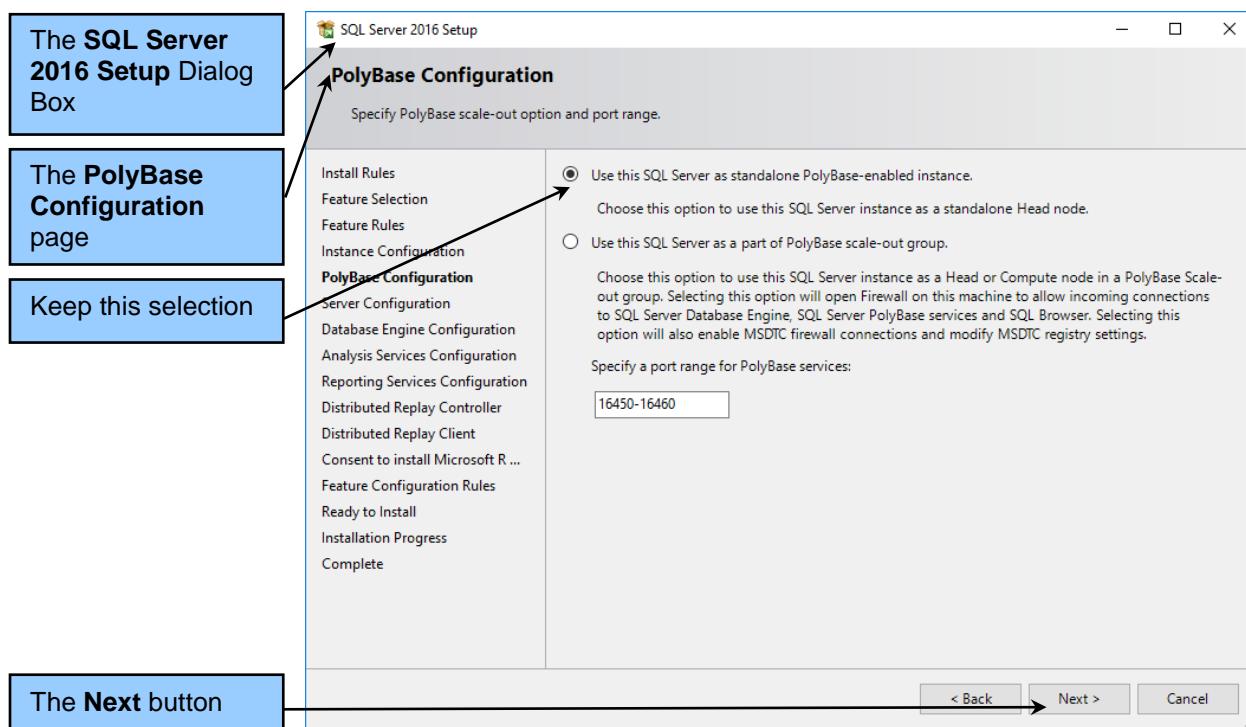


Figure A-8(l) — The SQL Server Installation Center Dialog Box PolyBase Configuration Page

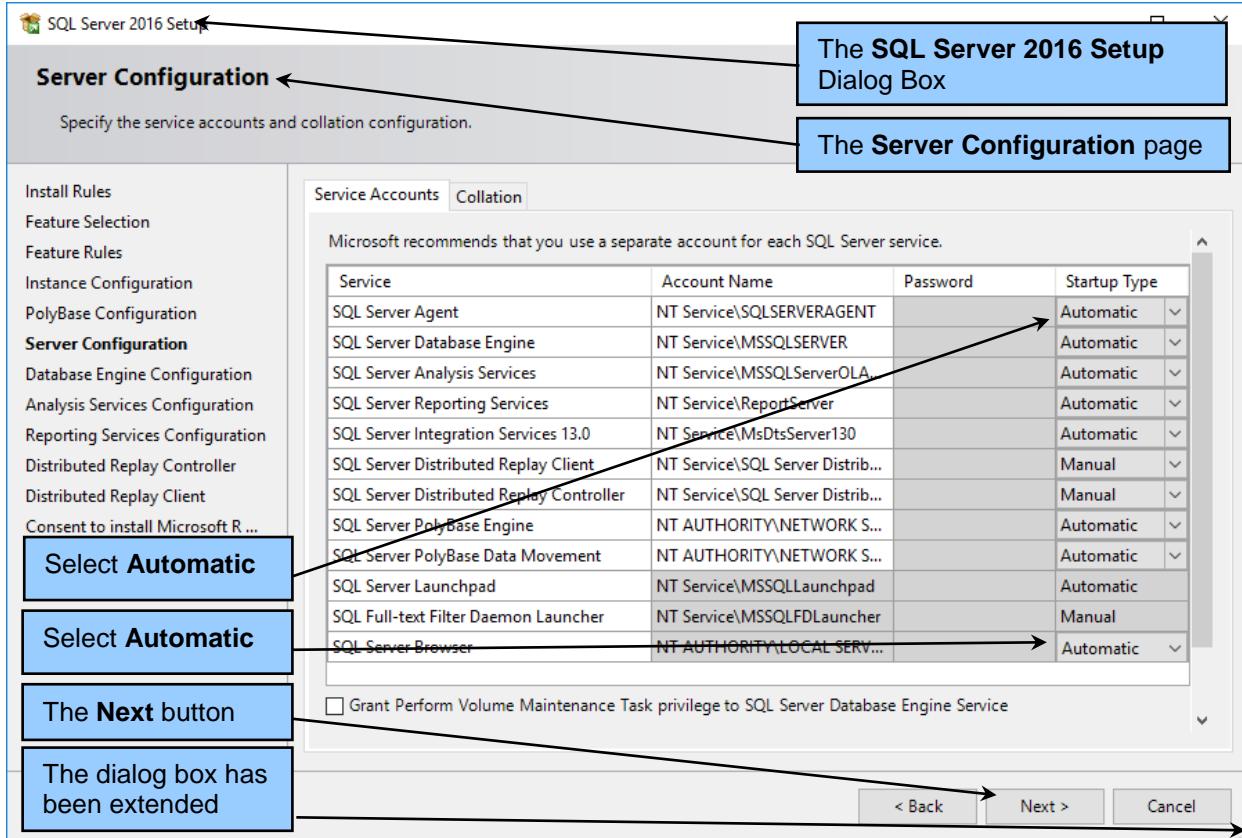


Figure A-8(m) — The SQL Server Installation Center Dialog Box Server Configuration Page

16. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Reporting Services Configuration** page is displayed, as shown in Figure A-8(p). The settings are correct as shown. (Note: For SQL Server 2016 Express, only the SQL Server 2016 Express Advanced edition supports Reporting Services, and this page will only appear in a setup for SQL Server 2016 Express Advanced.)
17. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Distributed Replay Controller** page is displayed, as shown in Figure A-8(q). Click the **Add Current User** button to give yourself administrative permissions for Analysis Services. *This page is displayed only if it is supported by the version of SQL Server 2016 you are installing.*
18. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Distributed Replay Client** page is displayed, as shown in Figure A-8(r). The settings on this page are correct. *This page is displayed only if it is supported by the version of SQL Server 2016 you are installing.*
19. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Consent to install Microsoft R Open** page is displayed, as shown in Figure A-8(s). Microsoft R Open is a statistical package that is used for data analysis. This page is displayed only if it is supported by the version of SQL Server 2016 you are installing. If this page is displayed, click the **Accept** button. All the other settings are correct.

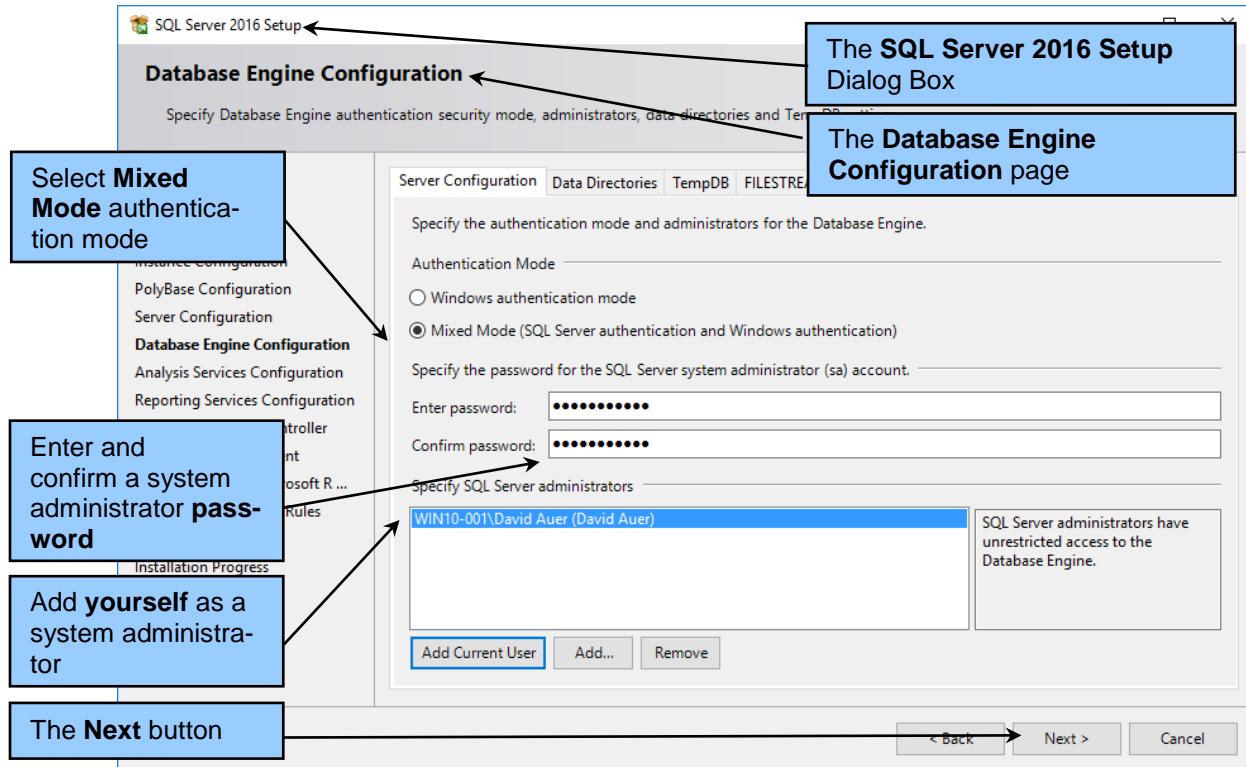


Figure A-8(n) — The SQL Server Installation Center Dialog Box Database Engine Configuration Page

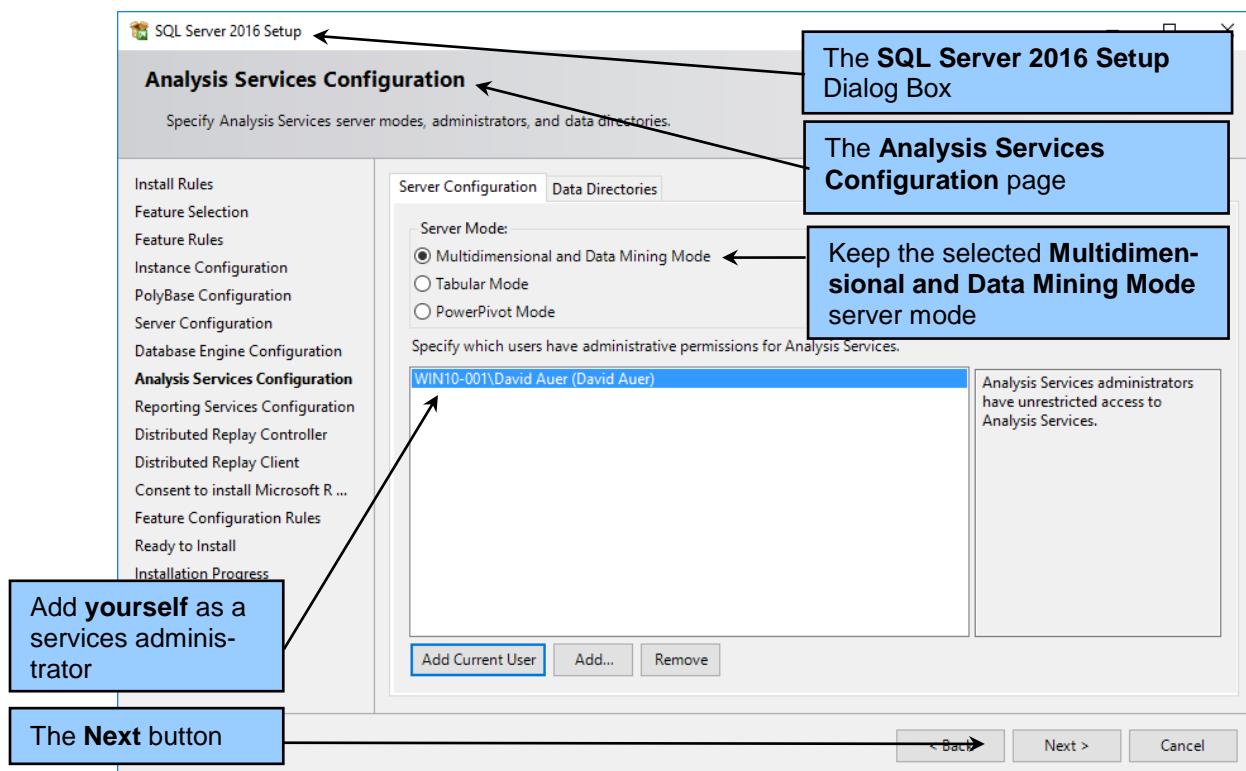


Figure A-8(o) — The SQL Server Installation Center Dialog Box Analysis Services Configuration Page

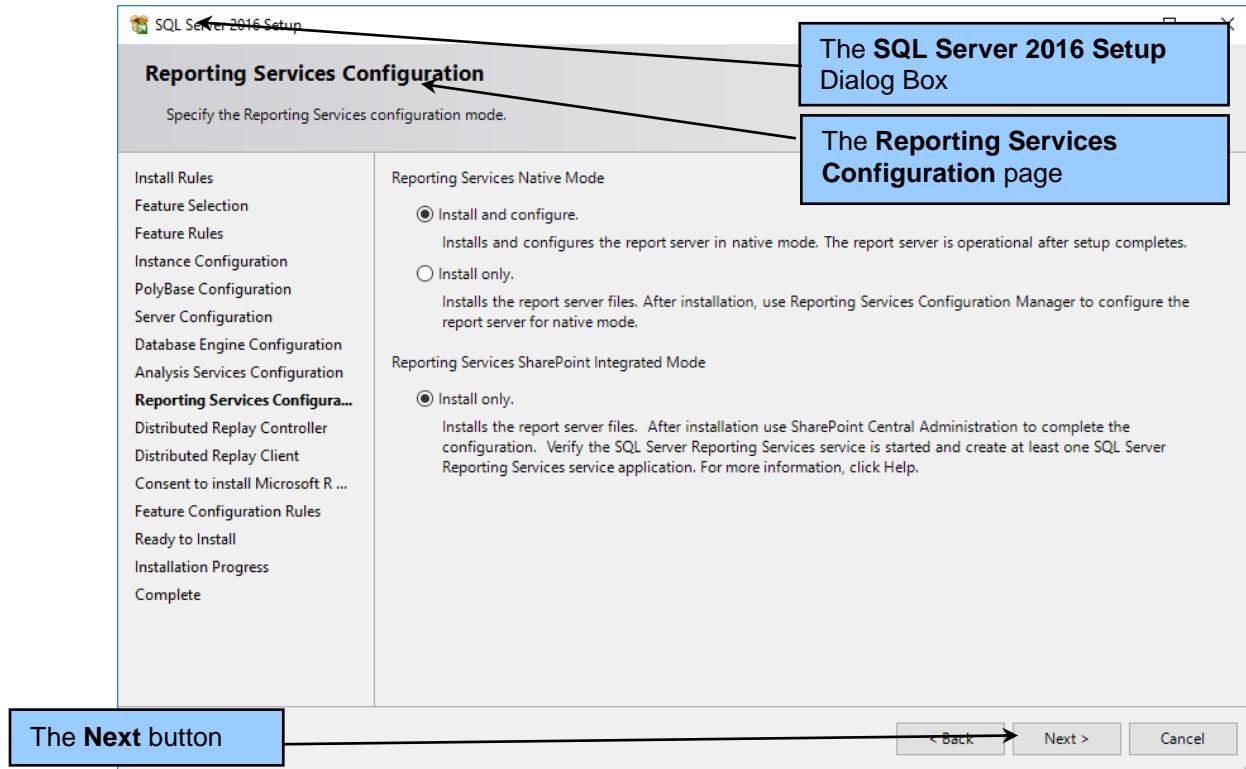


Figure A-8(p) — The SQL Server Installation Center Dialog Box Reporting Services Configuration Page

20. Click the **Next** button. The installation works through the process until the SQL Server 2016 Setup **Feature Configuration Rules** page is displayed, as shown in Figure A-8(t). *This page is displayed only if there are rules that did not pass.* To see this page, press the *Back* button, and then the *View Details* button. All rules should pass their tests.
21. If the *Feature Configuration Rules* page is displayed, click the **Next** button. The **Ready to Install** page is displayed, as shown in Figure A-8(u). This page summarizes the installation and displays the installation settings so that you can check to be sure they are correct. The settings shown here are correct for our installation process.
22. Click the **Install** button. SQL Server 2016 is installed (this will take a while, so be patient), as shown in Figure A-8(v).
23. When the installation is complete, the **Complete** page is displayed, as shown in Figure A-8(w).
24. Click the **Close** button to close the SQL Server 2016 Setup dialog box
25. **Do NOT close the SQL Server Installation Center Dialog Box at this time.** We will need it to install the administrative utilities we use with SQL Server 2016.

After you have installed SQL Server 2016, you should check for the latest service packs and patches using Windows Update to make sure your installation is as secure as possible. In particular, download and install the **Critical update for SQL Server 2016 MSVCRT prerequisites** from <https://support.microsoft.com/en-us/kb/3164398>. Other patches will be automatically installed by the Windows 10 Windows Update feature.

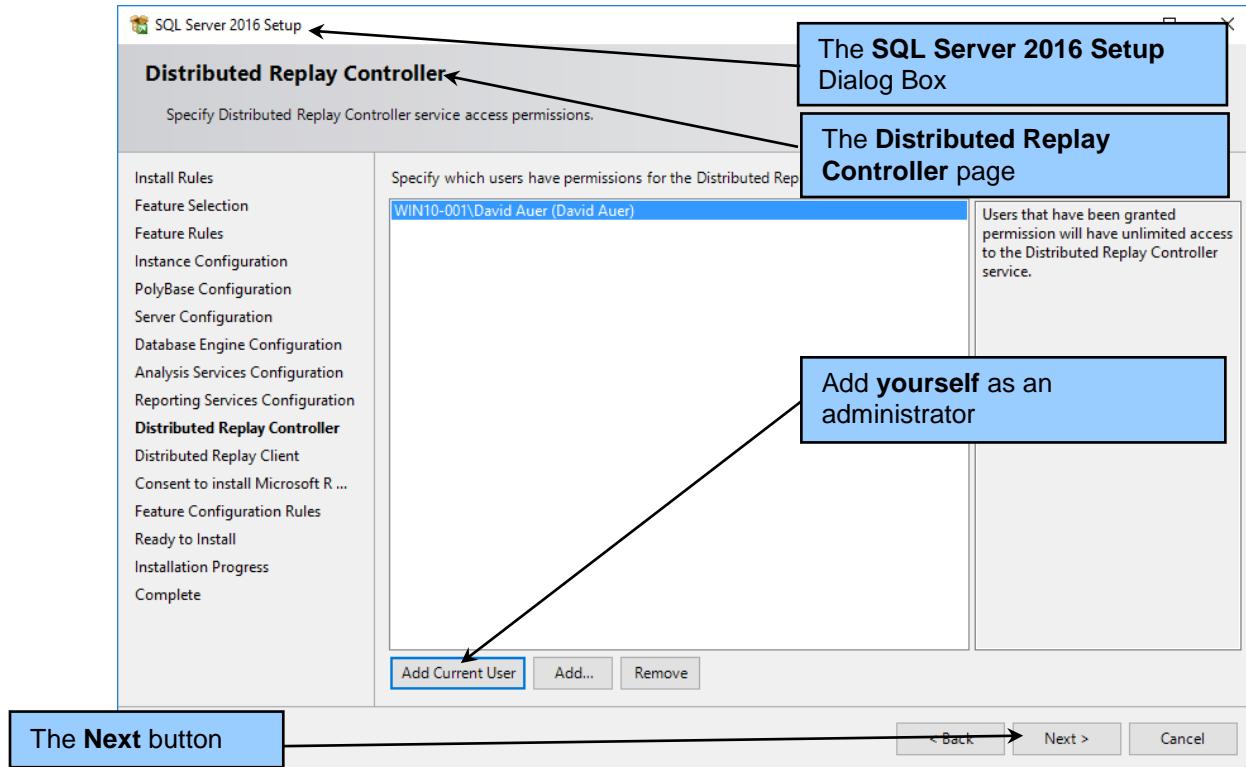


Figure A-8(q) — The SQL Server Installation Center Dialog Box Distributed Replay Controller Page

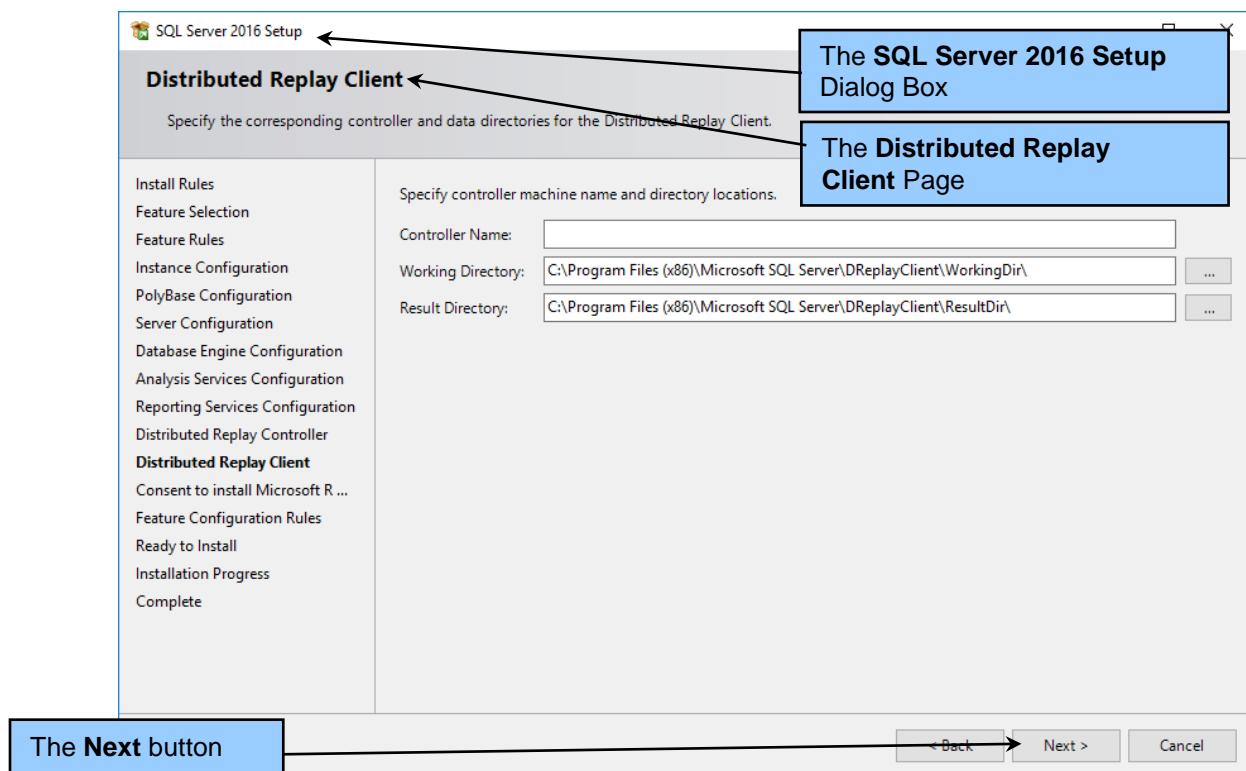


Figure A-8(r) — The SQL Server Installation Center Dialog Box Distributed Replay Client Page

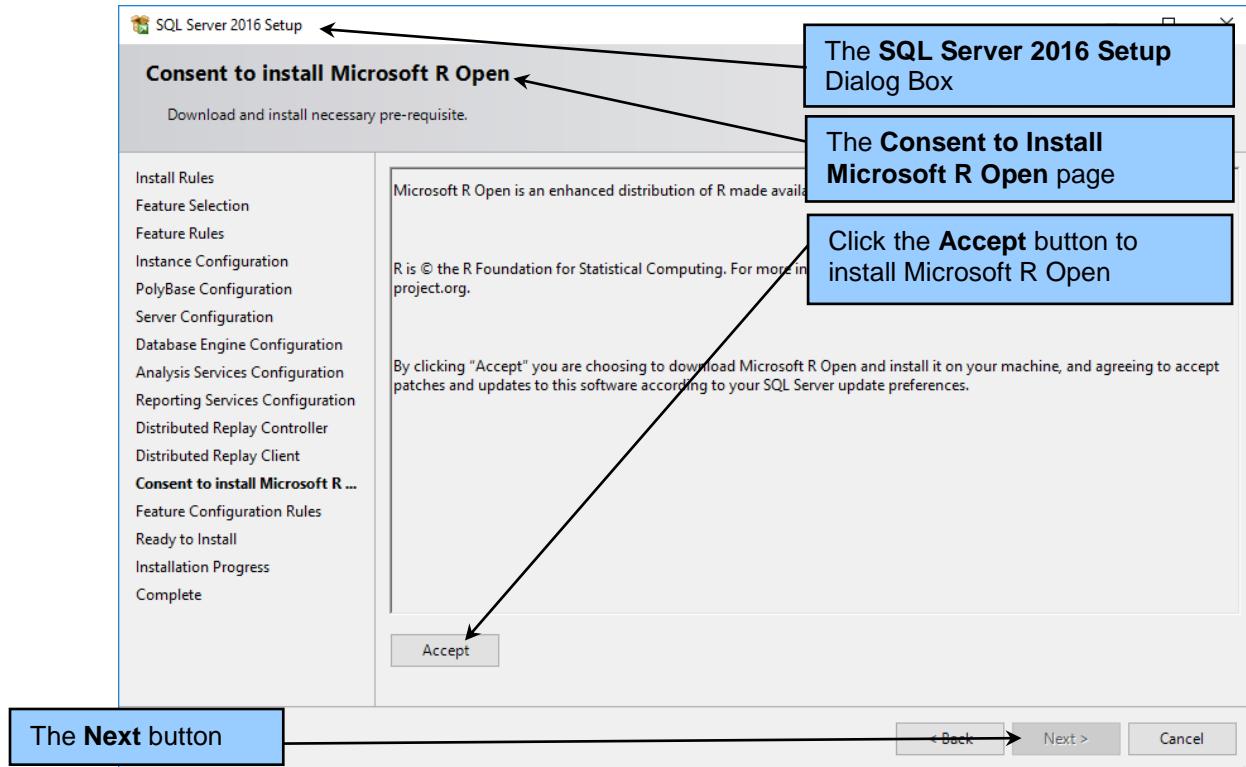


Figure A-8(s) — The SQL Server Installation Center Dialog Box Consent to install Microsoft R Open Page

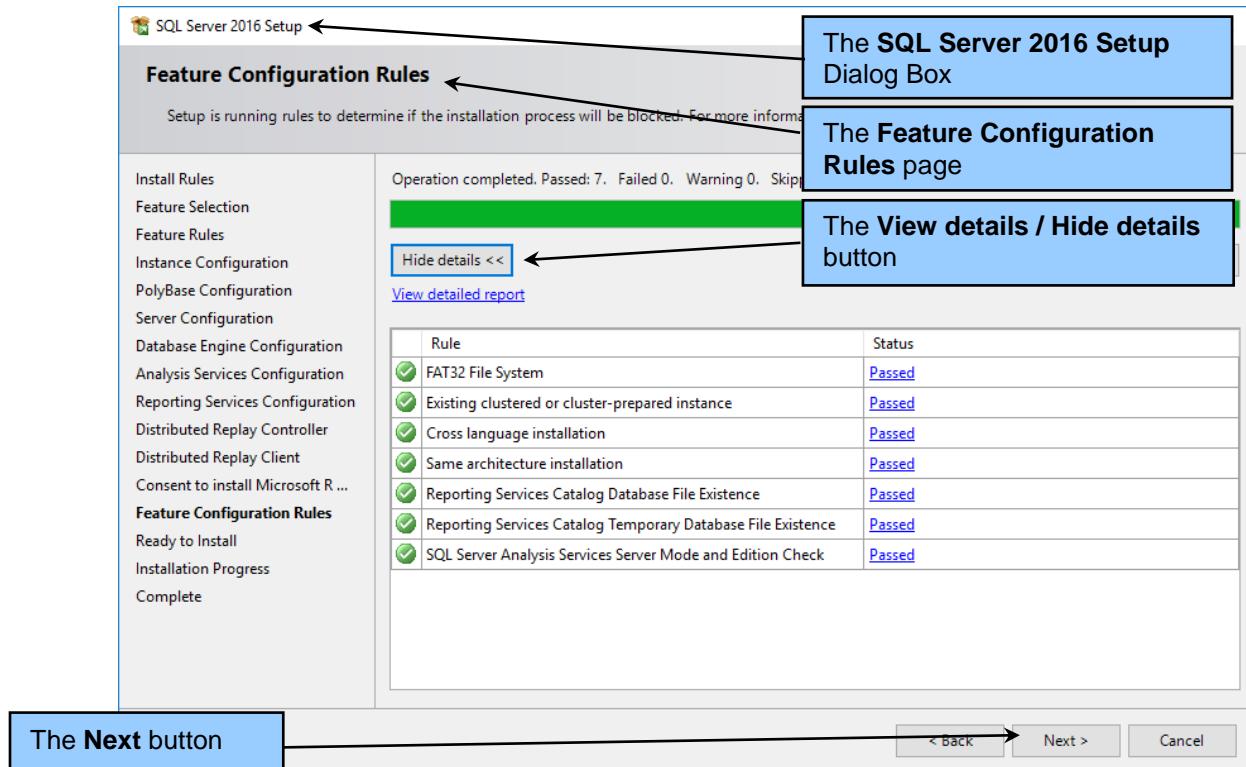


Figure A-8(t) — The SQL Server Installation Center Dialog Box Feature Configuration Rules Page

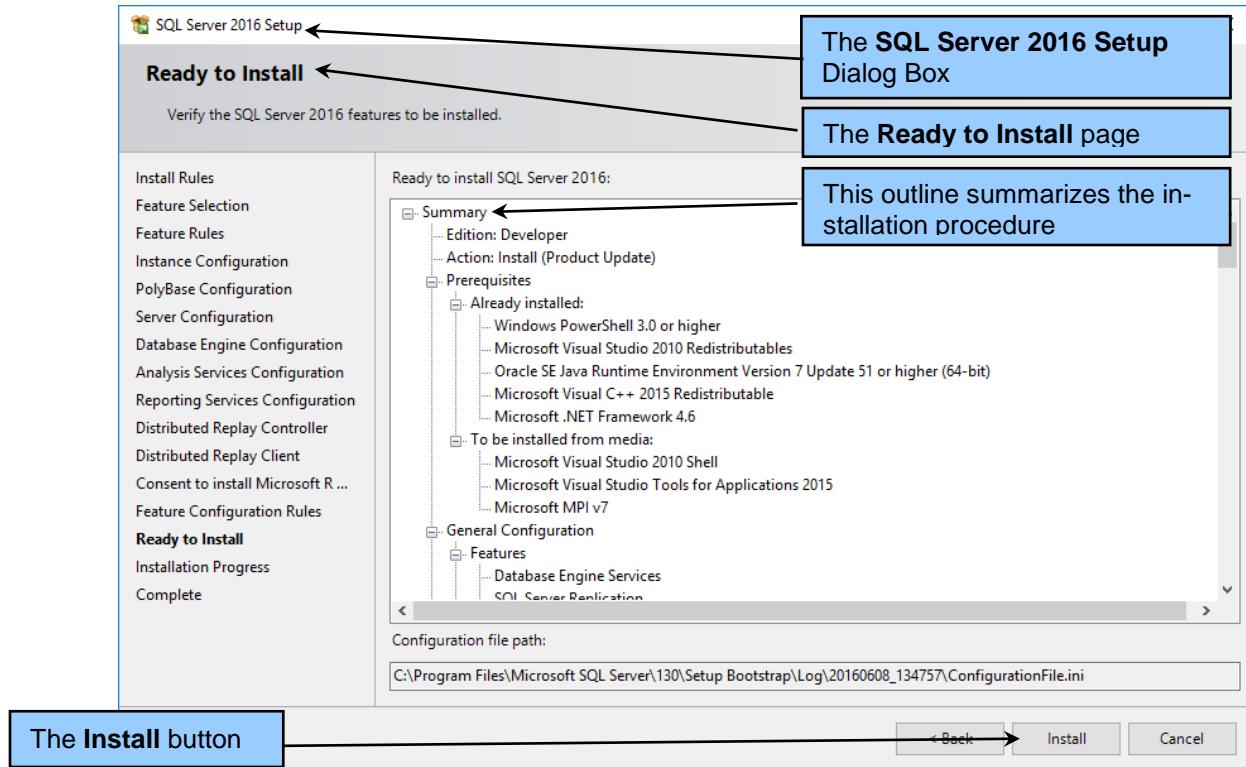


Figure A-8(u) — The SQL Server Installation Center Dialog Box Ready to Install Page

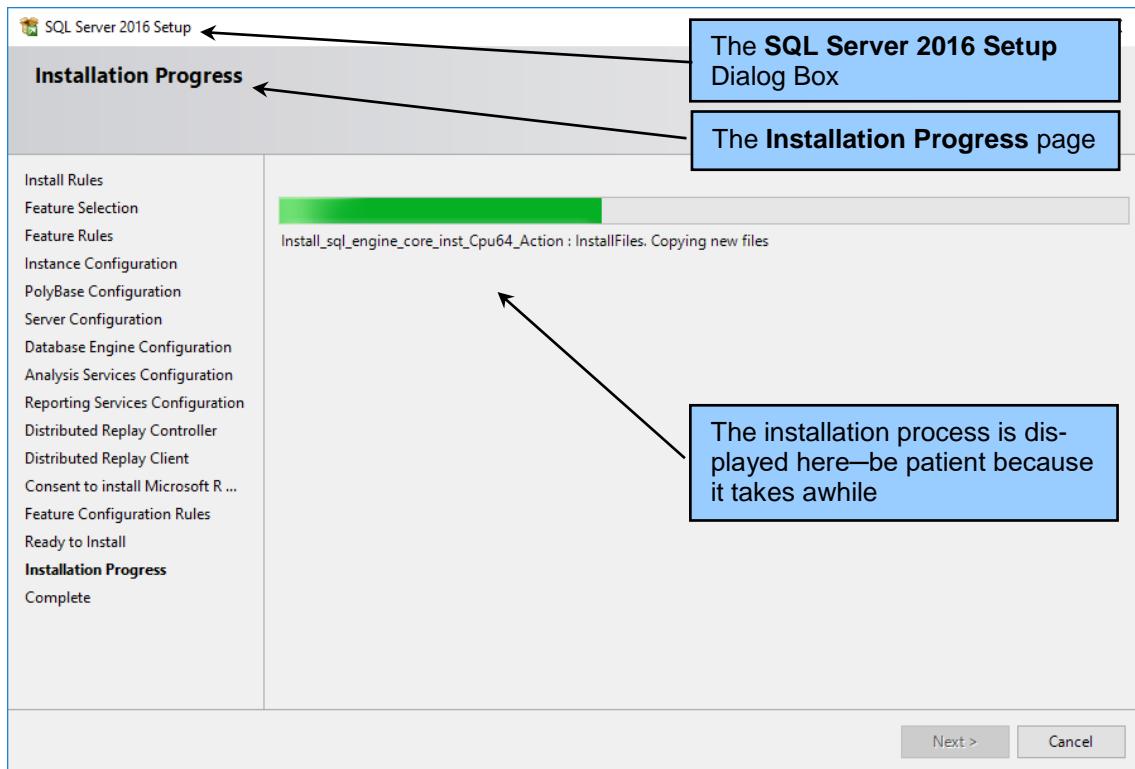


Figure A-8(v) — The SQL Server Installation Center Dialog Box Installation Progress Page

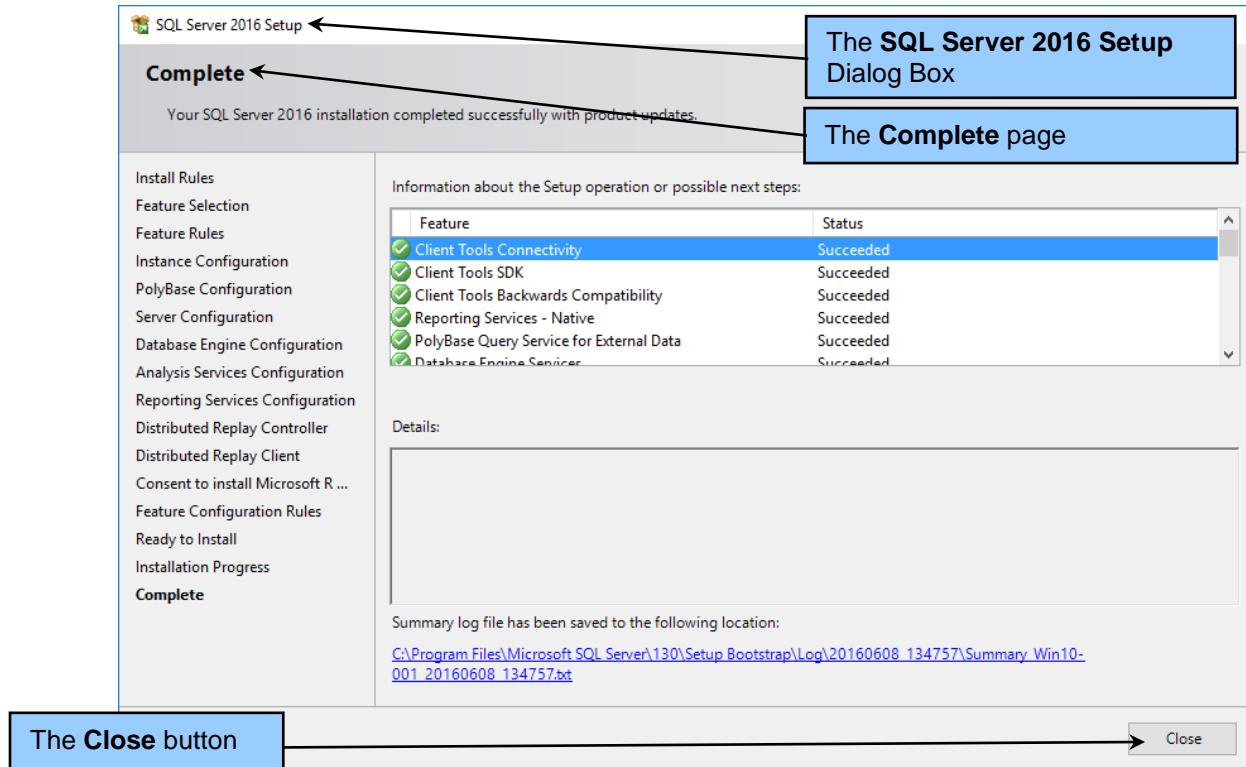


Figure A-8(w) — The SQL Server Installation Center Dialog Box Complete Page

How Do I Install the Microsoft SQL Server Management Studio?

Microsoft SQL Server 2016 provides both command-line utilities and graphical user interface (GUI) utilities for use with the Microsoft SQL Server DBMS. While some DBAs prefer command-line utilities, the current GUI utilities are very comprehensive and user friendly, and after introducing both types, we will use only the GUI utilities for the remainder of this appendix.

These utilities must be installed separately from Microsoft SQL Server 2016 itself (this is new with Microsoft SQL Server 2016 – in prior versions, these utilities were installed as part of the SQL Server engine installation). Figure A-8 illustrates the installation process of Microsoft SQL Server on a Windows operating system. Figure A-8(a) shows the *SQL Server Installation Center*, which is the first screen to appear when the installation process is begun. Here we have clicked on the **Installation** button to display the *Installation* screen.

Installing the Microsoft SQL Server 2016 Management Studio

1. To start the actual installation process, click on **Install SQL Server Management Tools** button in the SQL Server Installation Center, as shown in Figure A-9(a). The Microsoft Developer Network Download SQL Server Management Studio (SSMS) Web page is displayed, as shown in Figure A-9(b).

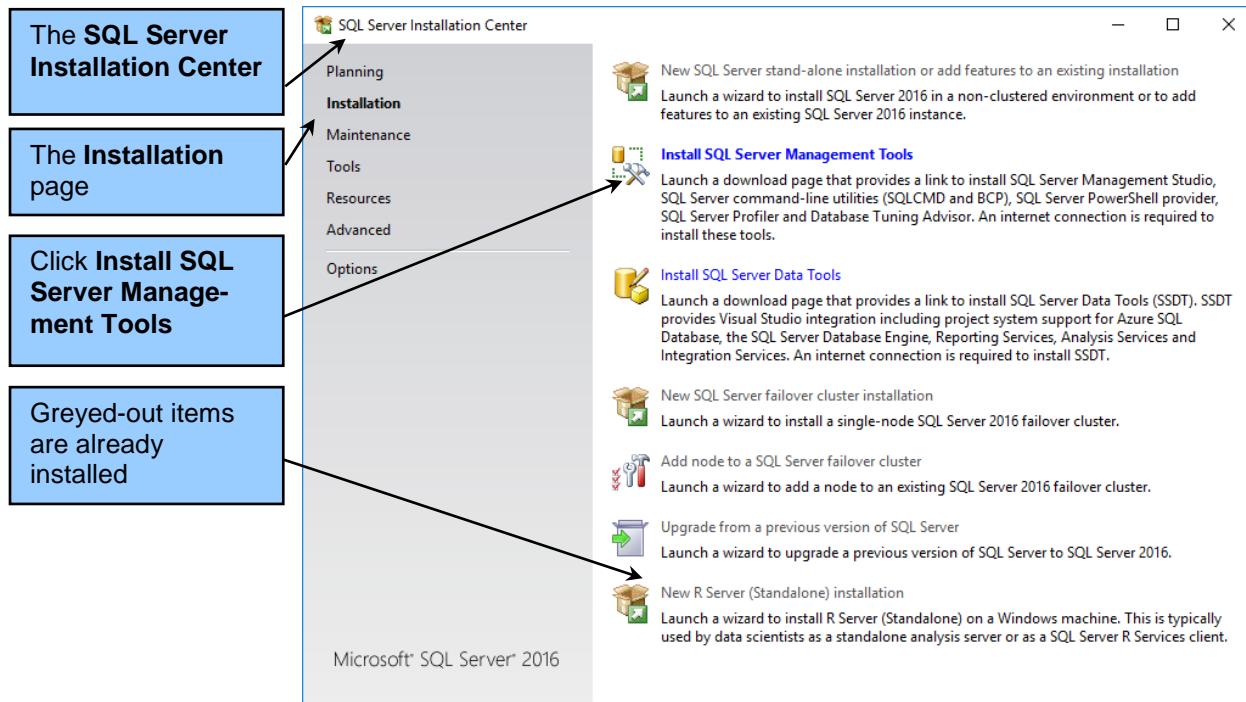


Figure A-9(a) — The SQL Server Installation Center Dialog Box Installation Page

Download SQL Server N × +

msdn.microsoft.com/en-us/library/mt238290.aspx

Microsoft | Developer Network

Downloads ▾ Programs ▾ Community ▾ Documentation ▾

... > SQL Server > SQL Server Tools > SQL Server Management Tools (including SSMS) ▾

Download SQL Server Management Studio (SSMS)

Updated: December 5, 2016

SQL Server Management Studio (SSMS) is an integrated environment for accessing, configuring, managing, administering, and developing all components of SQL Server. SSMS combines a broad group of graphical tools with a number of rich script editors to provide developers and administrators of all skill levels access to SQL Server. This release features improved compatibility with previous versions of SQL Server, a stand-alone web installer, and toast notifications within SSMS when new releases become available.

Download SQL Server Management Studio (SSMS)	
Download SQL Server Management Studio (16.5.1)	Current GA release for production use.
Download SQL Server Management Studio (17.0 RC1)	Includes support for SQL Server vNext CTP1, and works side-by-side with 16.x, but not recommended for production use.

Print Export (O) Share

IN THIS ARTICLE

- SQL Server Management Studio Changelog
- Previous releases Feedback See Also

Figure A-9(b) — The Download SQL Server Management Studio (SSMS) Web Page

2. Click the **Close** button to close the *Was this page helpful* message, and then click the **Download SQL Server Management Studio ({version})** (we are downloading version 16.5.1) link. Click the **Save** button to save the downloaded file to the Downloads folder
3. The **SSMS-Setup-ENU.exe** file is downloaded to the **This PC | Downloads** folder. When the download is complete, click the **Close** button in the message bar in the Web browser.
 - Note that the Download SQL Server Management Studio (SSMS) Web page contains additional information about using SSMS. You may want to save the Web page to your Favorites list for future reference.
4. Click the Web browser **Close** button to close the Web browser.
5. Open File Explorer and browse to the **This PC | Downloads** folder, as shown in Figure A-9(c). Right-click the **SMSS-Setup-ENU.exe** file to display the shortcut menu, and then click the **Run as administrator** command. When the User Account Control dialog box for this file is displayed, click the **Yes** button.
6. The **Microsoft SQL Server Management Studio installation dialog box** is opened, with the **Welcome. Click “Install” to begin.** page is displayed, as shown in Figure A-9(d). Click the **Install** button to begin the installation process.
7. The installation process is run (this takes some time, and various messages are displayed to show the installation progress – be patient), and when the installation is complete, the **Setup Completed** page is displayed, as shown in Figure A-9(e). Click the **Close** button to complete the install the installation process.
8. Close the **SQL Server Installation Center Dialog Box**.

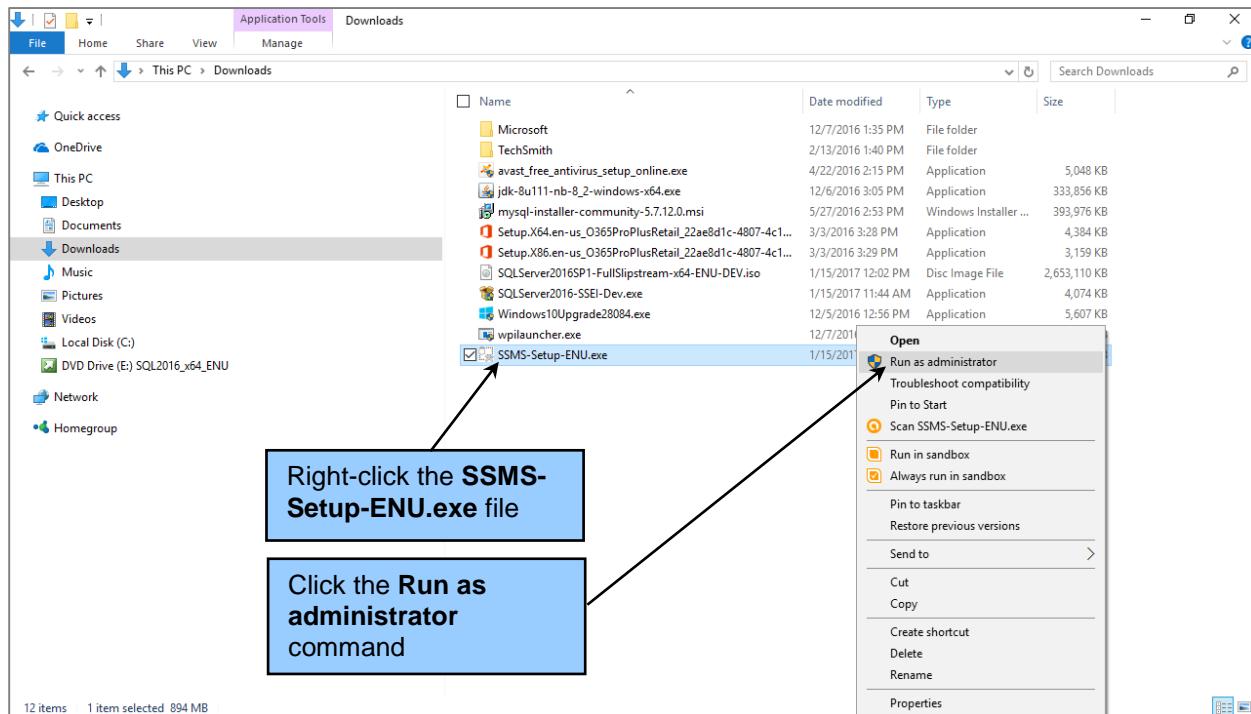


Figure A-9(c) — The SMSS-Setup-ENU.exe File and Shortcut Menu

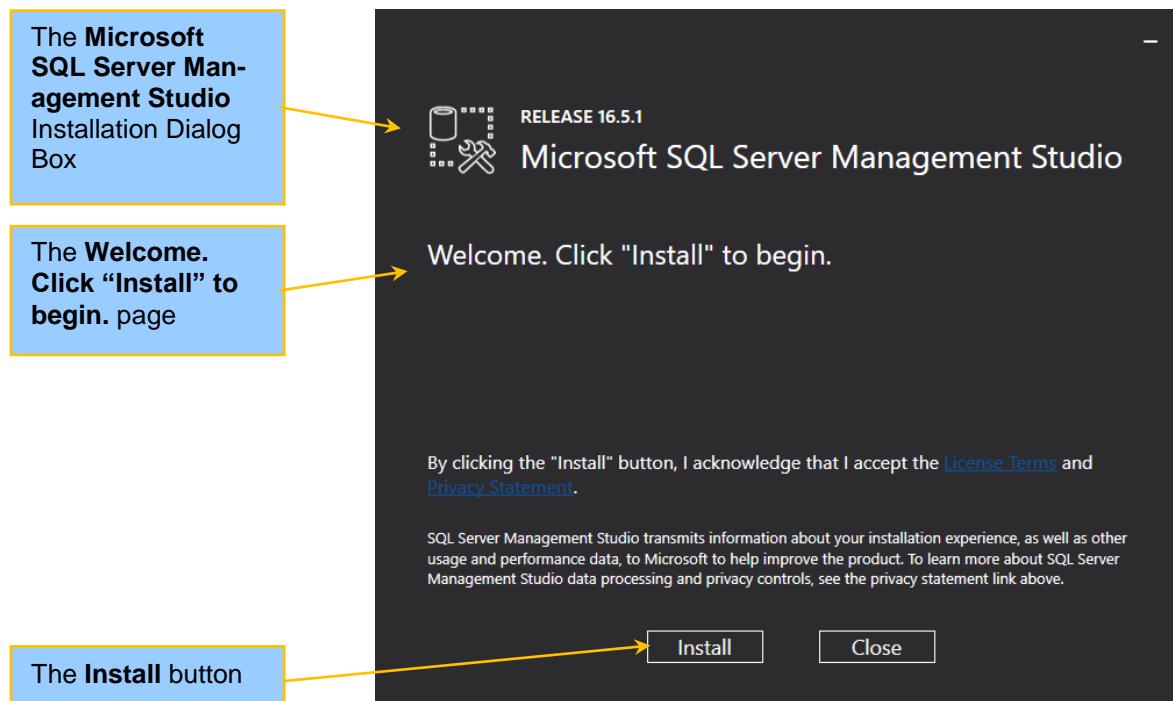


Figure A-9(d) — The Microsoft SQL Server Management Studio Installation Dialog Box Welcome Page

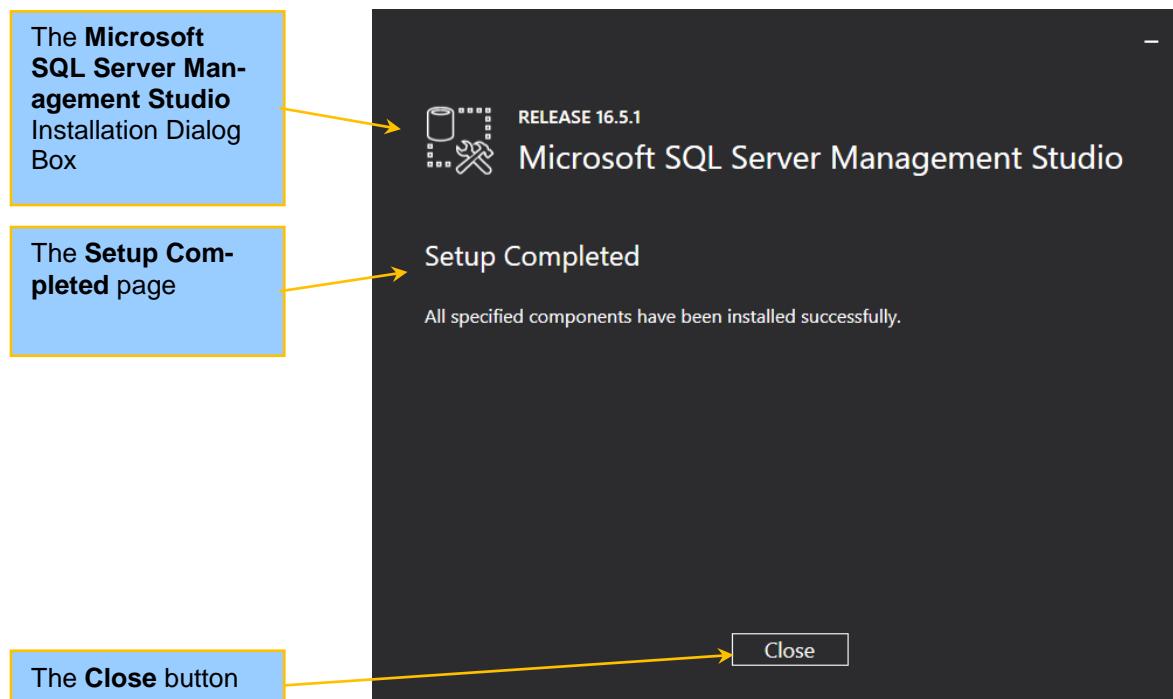


Figure A-9(e) — The Microsoft SQL Server Management Studio Installation Dialog Box Setup Complete Page

Starting the Microsoft SQL Server Management Studio

This appendix uses SQL Server 2016 – early versions of Microsoft SQL Server function similarly. To start working with SQL Server 2016 in Windows 10, click the **Windows Start** button, and then in the **All Apps** menu find **Microsoft SQL Server 2016 | Microsoft SQL Server Management Studio**. (In the Windows 7 operating system, select **Start | All Programs | Microsoft SQL Server 2016 | SQL Server Management Studio**.) The Microsoft SQL Server Management Studio **Connect to Server** dialog box appears, as shown in Figure A-10.

By The Way

You can pin the Microsoft SQL Server Management Studio Express program icon to either or both the Start Menu and the Taskbar. This makes it a lot easier to start the program if you are using it a lot.

To pin the program icon to either the Start Menu or the Taskbar, follow the command instructions above to locate the Microsoft SQL Server Management Studio program icon, and then **right-click** the program icon. A shortcut menu will be displayed. Select either **Pin to Start** or **More | Pin to Taskbar**.

If you use the default **Windows Authentication** method or **Mixed** authentication method (which we set up during Microsoft SQL Server 2016 installation), you will be authenticated using your current Windows username and password, so just click the **Connect button**. When you complete your login, the Microsoft SQL Server Management Studio window appears, as shown in Figure A-11.

In the Microsoft SQL Server Management Studio, you should have an **Object Explorer** window on the left side. The Object Explorer displays SQL Server 2016 objects (such as databases and tables) and the folders (with labels such as Databases and Security) that are used to organize the presentation of the objects. The folders are considered to be objects, and, therefore, the window contains an expandable set of objects. In Figure A-11, the **Databases folder** has been expanded to show that it contains a folder named System Databases (which holds important system databases created by SQL Server 2016) and a user-created database named Art_Course_Database (this database does not appear the first time you open the SQL Server Management Studio—we created it to illustrate a user-created database in the figure). Also note the databases named **ReportServer** and **ReportServerTempDB**—these are databases used by the **SQL Server 2016 Report Services** engine that is provided as part of the SQL Server 2016 package.

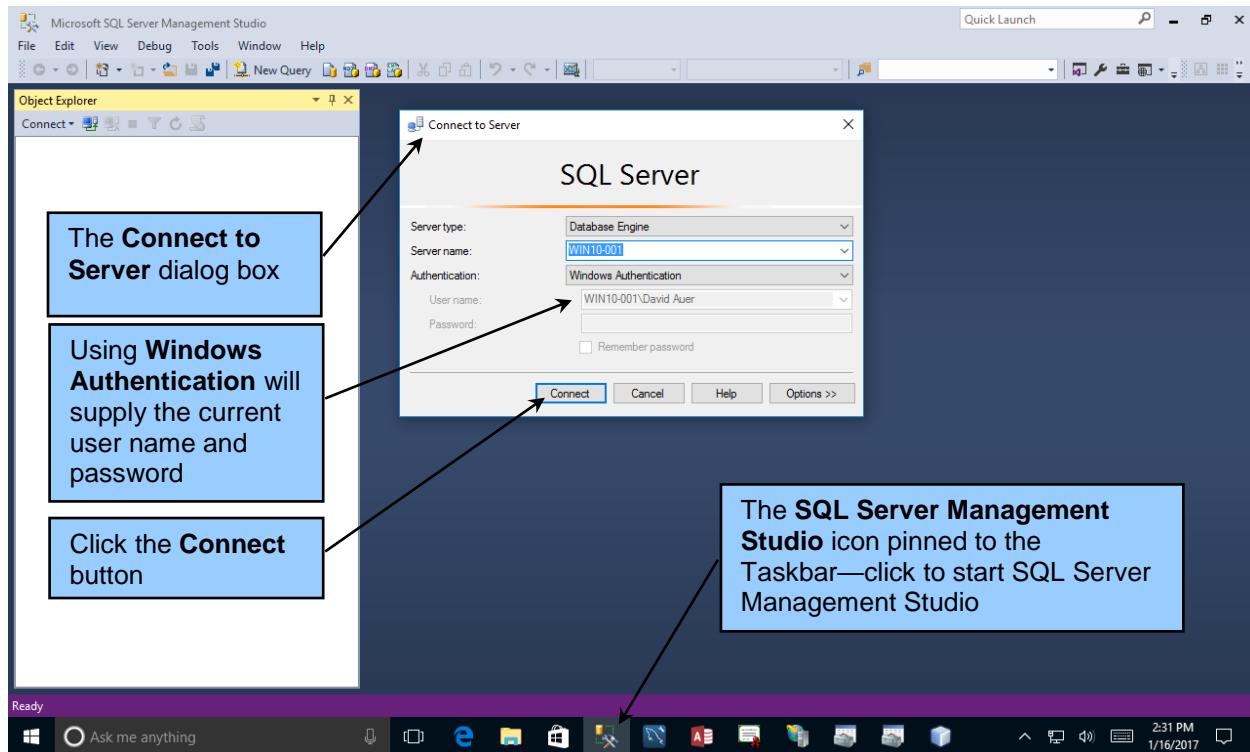


Figure A-10 — The Connect to Server Dialog Box

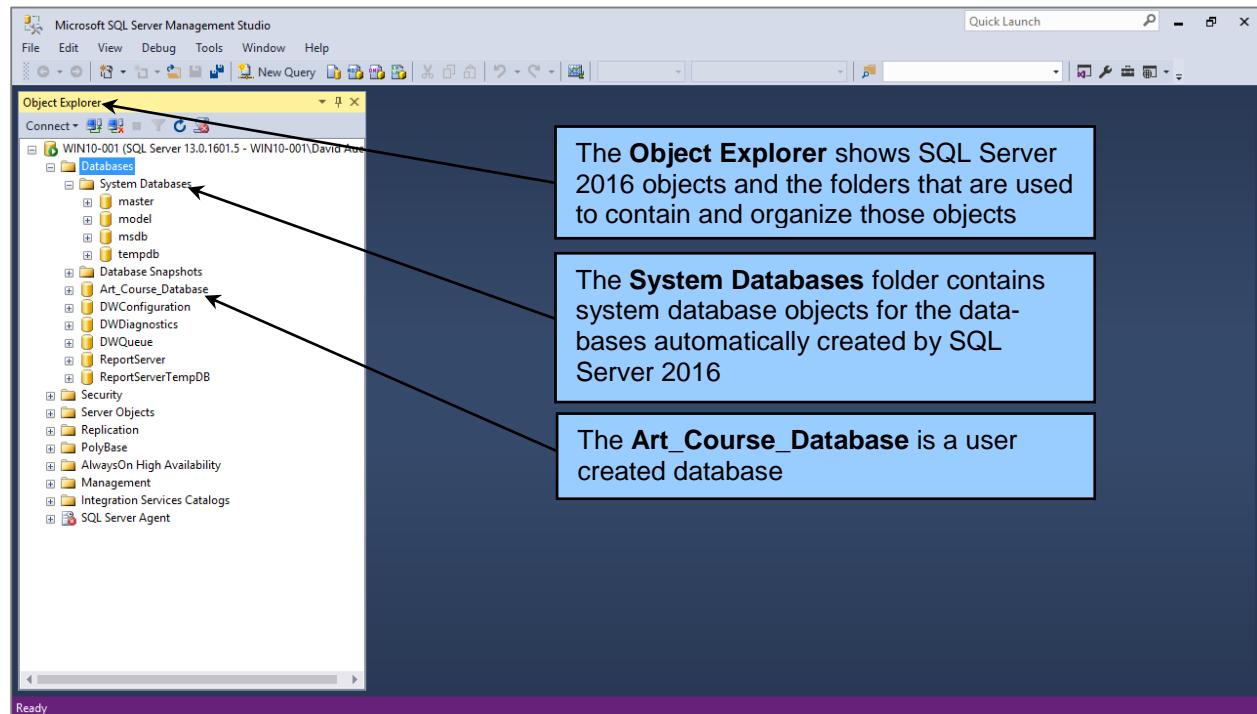
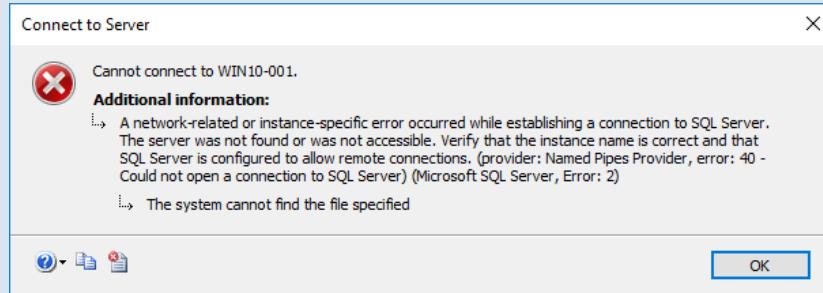


Figure A-11 — The Microsoft SQL Server Management Studio in Windows 10

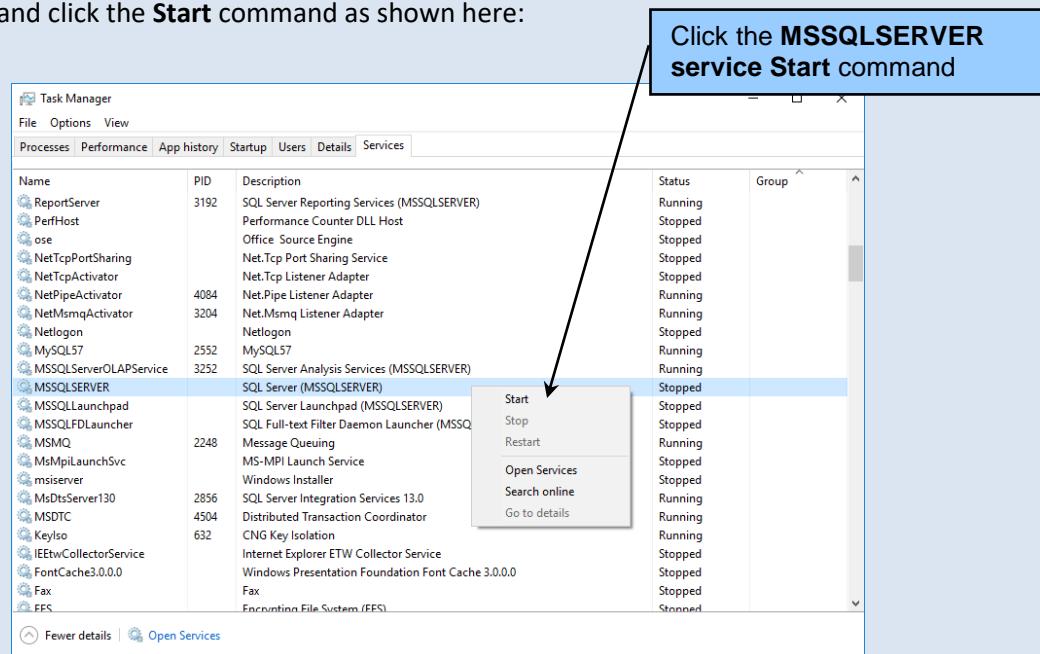
By The Way

We have repeatedly experienced a problem when starting the Microsoft SQL Server Management Studio, and when this occurs the following error message is displayed:



When this occurs, it indicates that, for some reason, the main SQL Server 2016 database engine has not started. Since the database engine is set to start automatically, we really don't understand why this error is happening.

To fix this problem if it occurs, click the **OK** button to close the error message, and then use the **Windows key + X** key combination on the keyboard to display a short cut menu. In the menu, click the **Task Manager** command to display the Task Manager. In Task Manager, click the **Services** tab and scroll down until you see the **MSSQLSERVER** service. Right-click the **MSSQLSERVER** service to bring up a shortcut menu and click the **Start** command as shown here:



Also start the **MSSQLLaunchpad** and **SQLSERVERAGENT** services. After these services are started, close the Task Manager and then log into Microsoft SQL Server 2016.

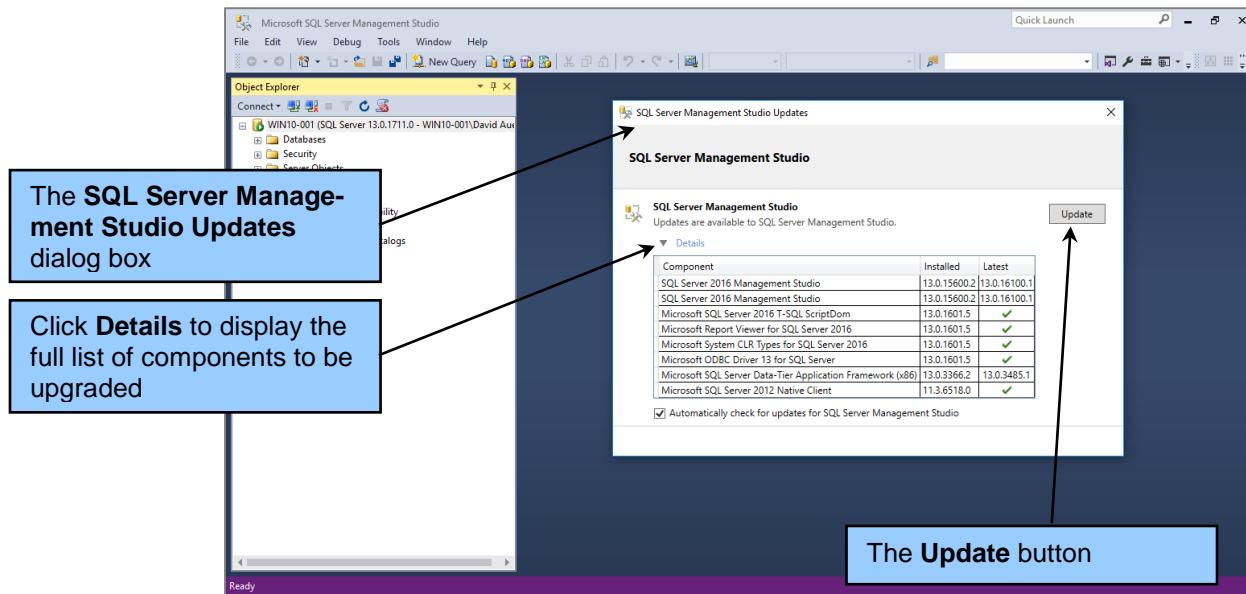


Figure A-12 — The Microsoft SQL Server Management Studio in Windows 10

Note that when you open the Microsoft SQL Server Management Studio, it will check for updates, notify you when an update is available, and ask if you want to get the update. Alternately, you can manually check for updates by using the **Tools | Check for Updates** command. If you click to accept the update, the SQL Server Management Studio Updates dialog box will be displayed, as shown in Figure A-12. When you click the Update button, you will be taken to the same Microsoft Developer Network Download SQL Server Management Studio (SSMS) Web page shown in Figure A-9(b). Download the update, close the SQL Server Management Studio, and install the updated version using the same steps previously discussed.

How Do I Create a New Database in Microsoft SQL Server 2016?

To create an SQL Server database, we use commands in the Microsoft SQL Server Management Studio.

Creating a New Database in Microsoft SQL Server 2016

1. To create a new database, right-click the **Databases** object in the Object Explorer to display the shortcut menu, as shown in Figure A-13.
2. In the shortcut menu, click the **New Database** command. The **New Database** dialog box appears, as shown in Figure A-14.
3. Type the name of the new database—in this example, **WP** (for Wedgewood Pacific)—in the Database Name text box and then click the **OK** button to create the new database.
4. After you create the database, the new database name appears in the Databases folder object. Right-click the **Databases** object in the Object Explorer to display the shortcut menu, and then

click the **Refresh** command. Figure A-15 shows the WP database folder object selected and expanded to show the objects within the WP database. It also shows the System Databases folder expanded.

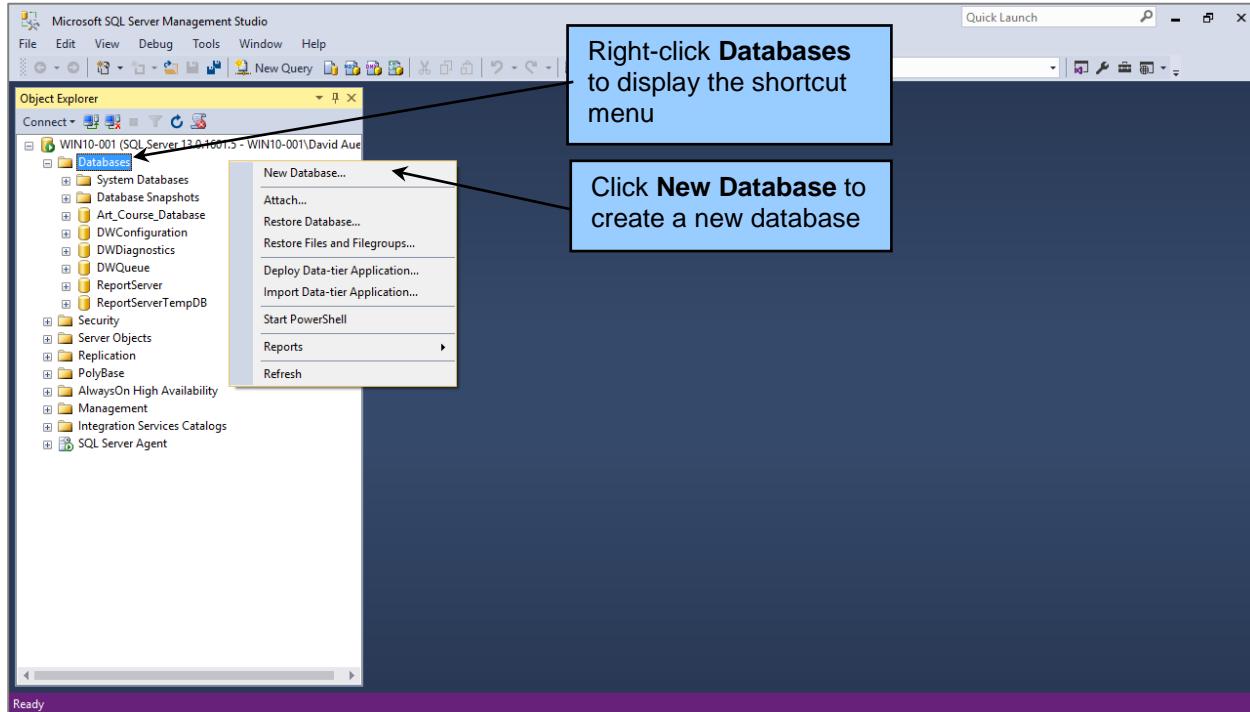


Figure A-13 — The New Database Command

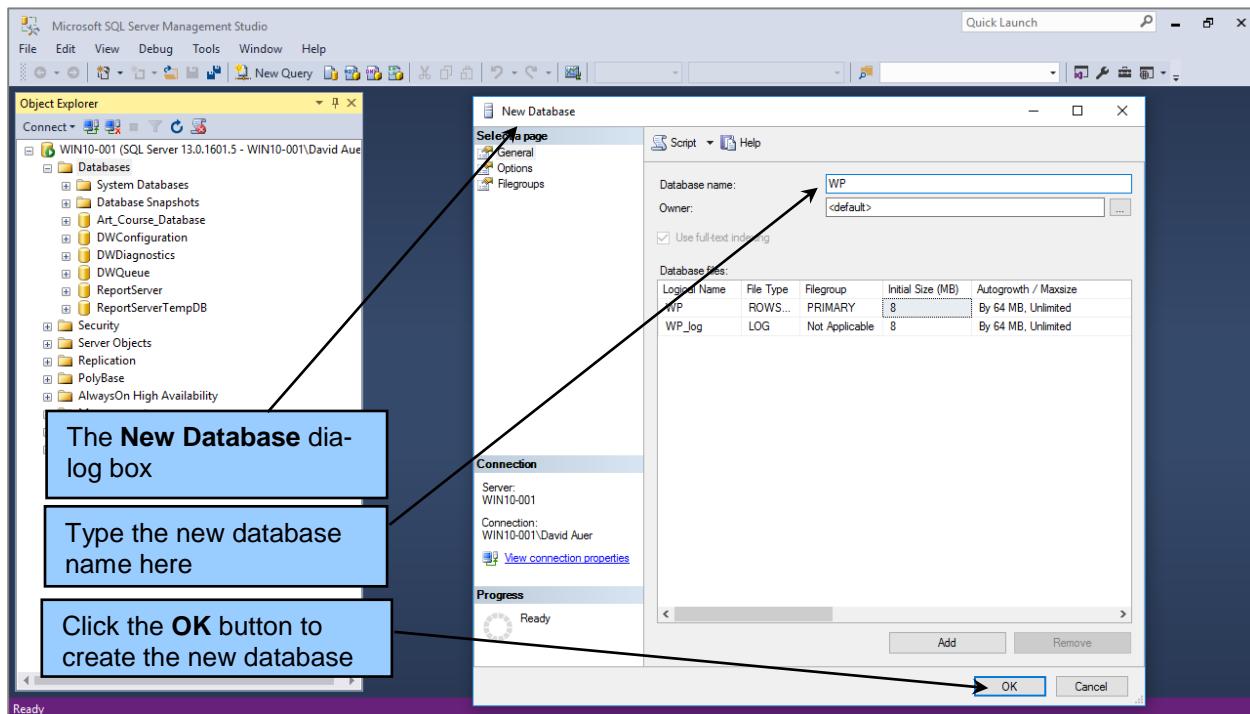
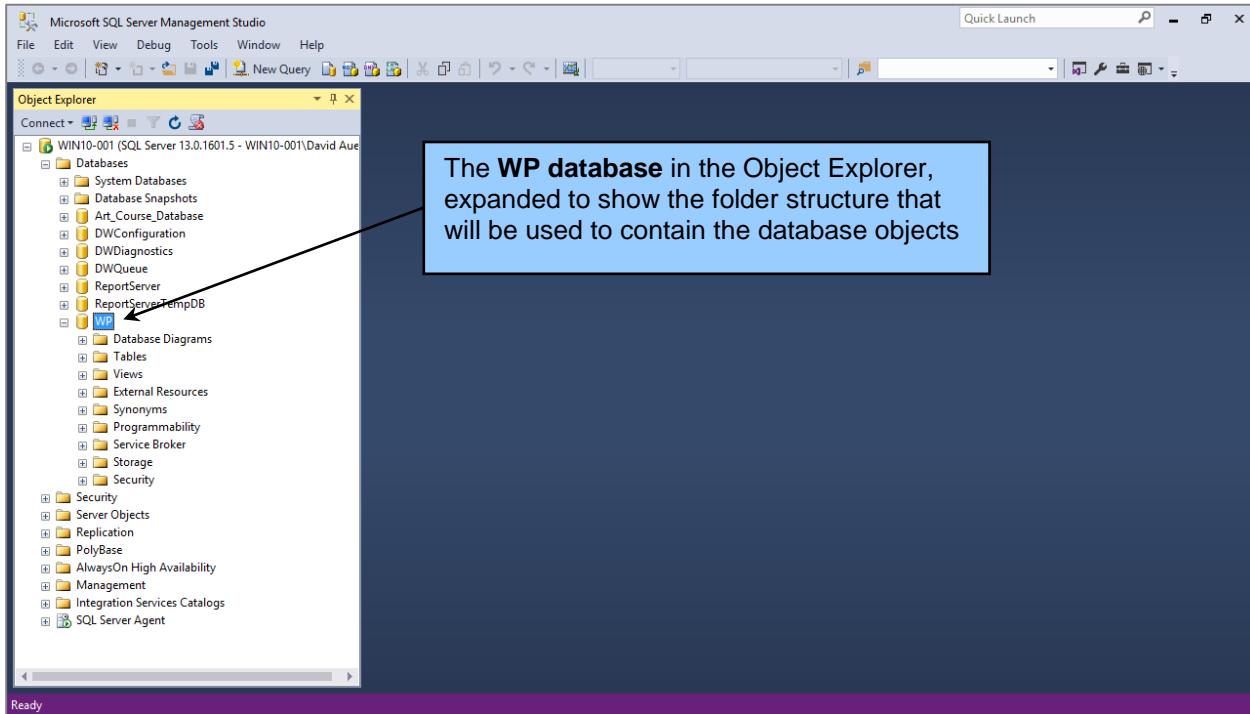


Figure A-14 — Naming the New Database**Figure A-15 — The New Database in the Object Explorer**

How Do I Use SQL Statements in SQL Server?

In SQL Server, SQL statements can be run individually or as part of a related group of SQL statements known as an **SQL script**. SQL scripts 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.

How Do I Work with SQL Scripts in SQL Server?

You can create scripts in the SQL Server Management Studio or in any ASCII text editor. In the Windows operating system, the Notepad text editor is a good choice if you're not using SQL Server Management Studio itself. Regardless of which text editor you use, save your scripts with the file extension ***.sql** so that SQL Server recognizes them. By default, SQL Server Management Studio looks for scripts in a special folder that is created in the user's **This PC | Documents** folder when the program is installed on a personal workstation. The folder structure created by the installation process is shown in Figure A-16, along with a *newly created* folder named **Projects** that we have created. SQL scripts should be stored in the **Projects** folder, in a subfolder named for the database.

By The Way

Using the Microsoft SQL Server Management Studio for Microsoft SQL Server 2016, we have to create the Projects folder, as shown in Figure A-16. In some previous versions of the Microsoft SQL Server Management Studio, the Projects folder is automatically created (along with the SQL Server Management Studio folder and the other subfolders).

If you are using an earlier version of Microsoft SQL Server such as Microsoft SQL Server 2008 R2, you find the Projects folder already created for you. You will still have to create a subfolder for each database.

How Do I Use SQL Commands to Create Table Structures?

The SQL statements to create the WP database are shown in Figure 3-7, which is repeated as Figure A-17 on the next page. These SQL Statements *are* SQL Server SQL commands, and we will use them here. We will also use the SQL Server Management Studio as our text editor.

To create a new document for our SQL script, click the New Query button shown in Figure A-18. A new tabbed document window, in this case named SQLQuery1.sql, is displayed to the right of the Object Browser. The **SQL Editor toolbar** is also displayed, which contains tools useful when editing text. One of

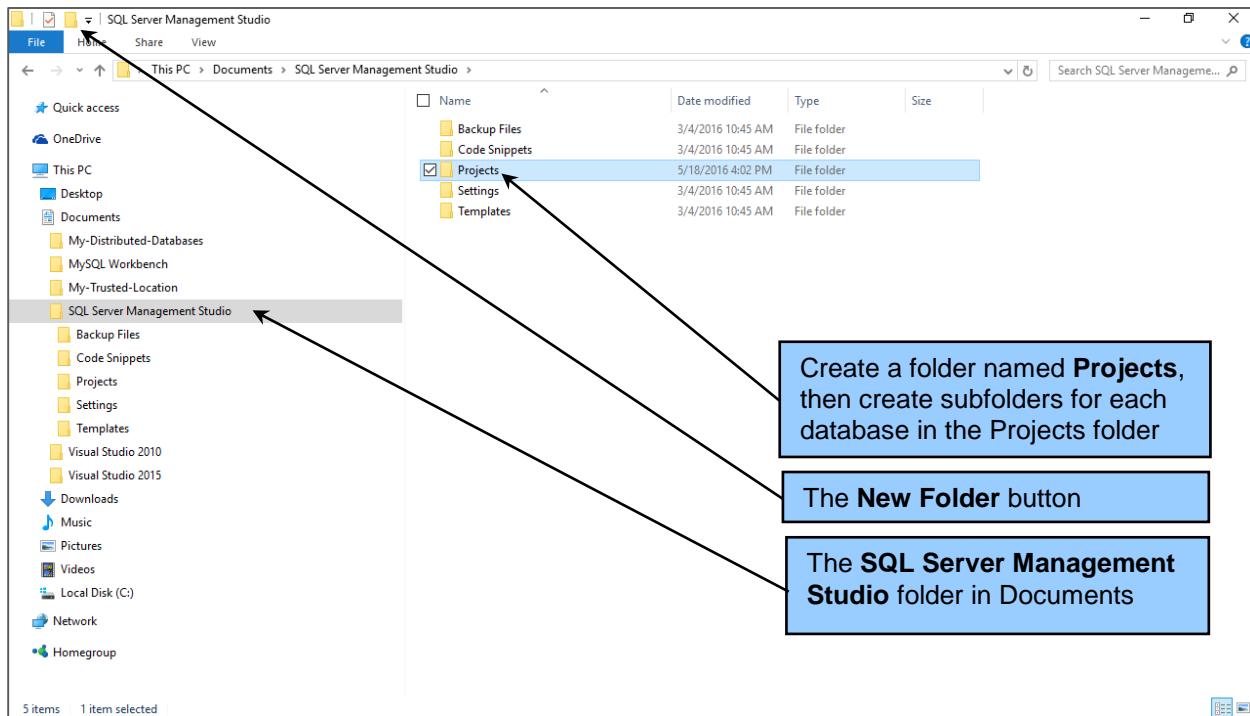


Figure A-16 — The Microsoft SQL Server Management Studio Folder in Windows Explorer

```
USE WP
GO

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

CREATE TABLE EMPLOYEE(
    EmployeeNumber Int NOT NULL IDENTITY (1, 1),
    FirstName Char(25) NOT NULL,
    LastName Char(25) NOT NULL,
    Department Char(35) NOT NULL DEFAULT 'Human Resources',
    Position Char(35) NULL,
    Supervisor Int NULL,
    OfficePhone Char(12) NULL,
    EmailAddress VarChar(100) NOT NULL UNIQUE,
    CONSTRAINT EMPLOYEE_PK PRIMARY KEY(EmployeeNumber),
    CONSTRAINT EMP_DEPART_FK FOREIGN KEY(Department)
        REFERENCES DEPARTMENT(DepartmentName)
        ON UPDATE CASCADE
);

CREATE TABLE PROJECT (
    ProjectID Int NOT NULL IDENTITY (1000, 100),
    ProjectName Char(50) NOT NULL,
    Department Char(35) NOT NULL,
    MaxHours Numeric(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 A-17 — The Figure 3-7 SQL Statements to Create the WP Database Tables

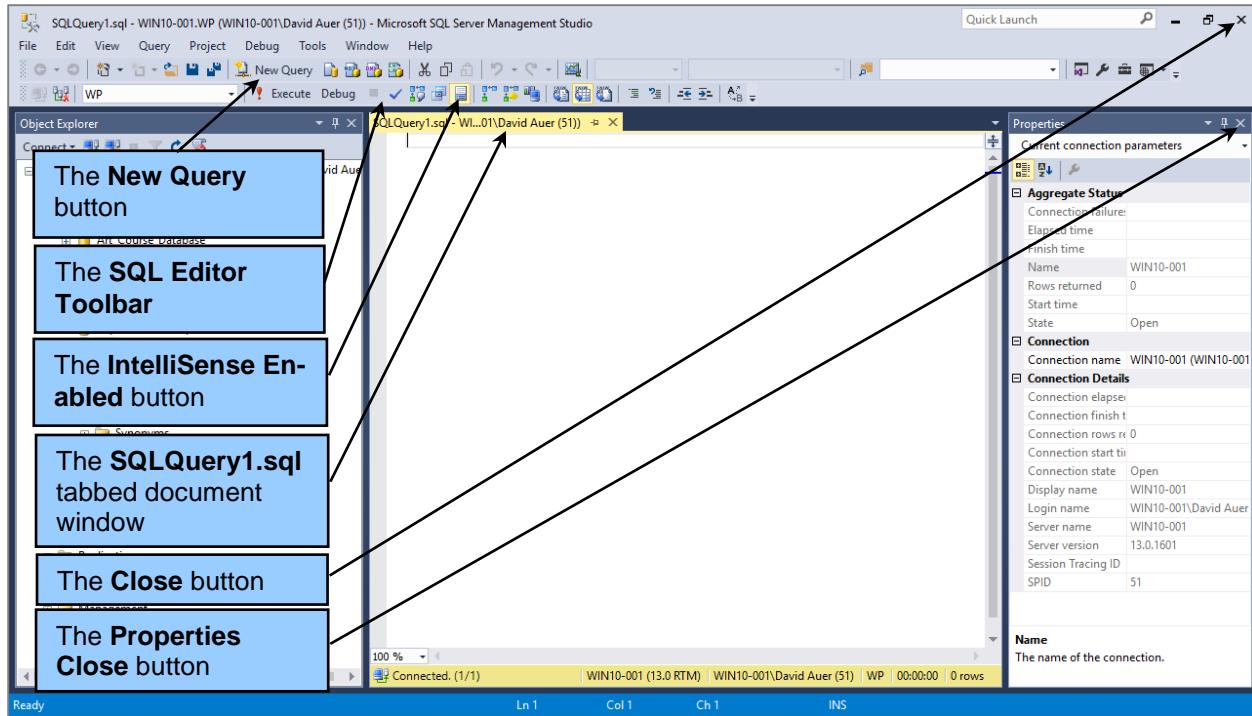


Figure A-18 — The SQLQuery1.sql Tabbed Document Window

these is the **IntelliSense Enabled button**, which may be useful to you as it provides some outlining and auto-completion capabilities. However, you may also find it more confusing to see the notations and auto-completion suggestions that are displayed (I know I often do!), and you can simply click the IntelliSense button to disable this feature. Also note the location of the **Close button**, used to close the tabbed document window, and the **Properties Close button**, used to close the Properties Window. At this time, use the Properties Close button to close the Properties window because we will not be using it.

Creating an SQL Script in Microsoft SQL Server 2016

1. In the tabbed document window, type in the SQL statements shown in Figure A-17. Be sure to use the same uppercase and lowercase letters shown in the figure, and also be sure to indent these statements as shown (this is done to help us read the statements easily). Note the color coding used in the SQL statements. When you are done, your work will look like the SQL statements shown in Figure A-19.
2. Click the **Save** button. The **Save File As** dialog box appears, as shown in Figure A-20. Browse to the Projects folder, and create a new folder named **WP-Database**.
3. Browse to the WP-Database folder, and then save the SQL script with the name **DBC-e08-MSSQL-WP-Create-Tables.sql** (you can use a shorter name if you prefer, such as just **WP-Create-Tables.sql**).
4. Figure A-21 shows the **Available Databases** drop-down list, which is used to select the **active database**—the one that the script will be run against. Select the WP database from the list.

5. Figure A-21 shows the **Parse** button, which is used to test the SQL code before actually running it. Click this button to test your SQL code:
- The **Parse** button, which is used to test the SQL code before actually running it.
 - The **Execute** button, which is used to actually run the SQL code in the document window.

Test your code until the parser doesn't find any errors, save the script again, and then execute the script. Figure A-22 shows the tables in the WP database.

By The Way

SQL Server is very sensitive about using SQL keywords as table or column names. Avoid keywords such as *name*, *date*, and *transaction*. Use modified versions of such words whenever possible—*ProjectName*, *StartDate*, *ItemTransaction*—or enclose the unmodified words in square brackets—[*Name*], [*Date*], [*Transaction*]—if you must use them.

How Do I Use SQL Statements to Insert Database Data?

The SQL statements needed to insert data into the tables you have created are the same as the ones shown in Figure 3-12, which are repeated in Figure A-23 on the next page. You also need to put this data set into a new script file, save it, *be sure the WP database is selected as the active database*, and then run it as an SQL script.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'WP' database is selected. A callout box labeled 'The Save button' points to the 'Save' icon in the toolbar. Another callout box labeled 'The SQL statements from Figure A-17—note the comments added as a header' points to the top of the query editor window, where a multi-line comment block is present. The query editor contains the following SQL code:

```
/*
 * Kroenke and Auer - Database Concepts (8th Edition) Chapter 03
 */
/*
 * Wedgewood Pacific (WP) Create Tables
 */
/*
 * These are the Microsoft SQL Server 2014/2016 SQL code solutions
 */
USE WP
GO

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

CREATE TABLE EMPLOYEE(
    EmployeeNumber   Int          NOT NULL IDENTITY (1, 1),
    FirstName       Char(25)      NOT NULL,
    LastName        Char(25)      NOT NULL,
    Department      Char(35)      NOT NULL DEFAULT 'Human Resources',
    Position        Char(35)      NULL,
    Supervisor      Int          NULL,
    OfficePhone     Char(12)      NULL,
    EmailAddress    VarChar(100)   NOT NULL UNIQUE,
    CONSTRAINT      EMPLOYEE_PK PRIMARY KEY(EmployeeNumber),
    CONSTRAINT      EMP_DEPART_FK FOREIGN KEY(Department)
                    REFERENCES DEPARTMENT(DepartmentName)
);
```

Figure A-19 — The SQL Script to Create the WP Database Tables

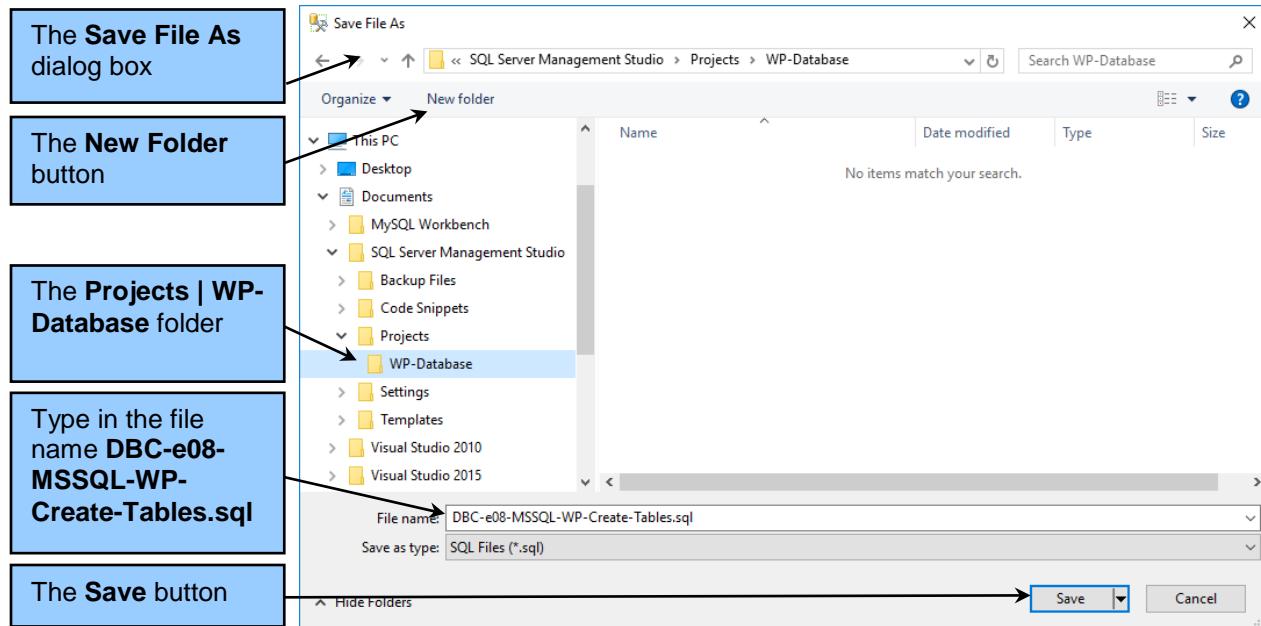


Figure A-20 — The Save File As Dialog Box

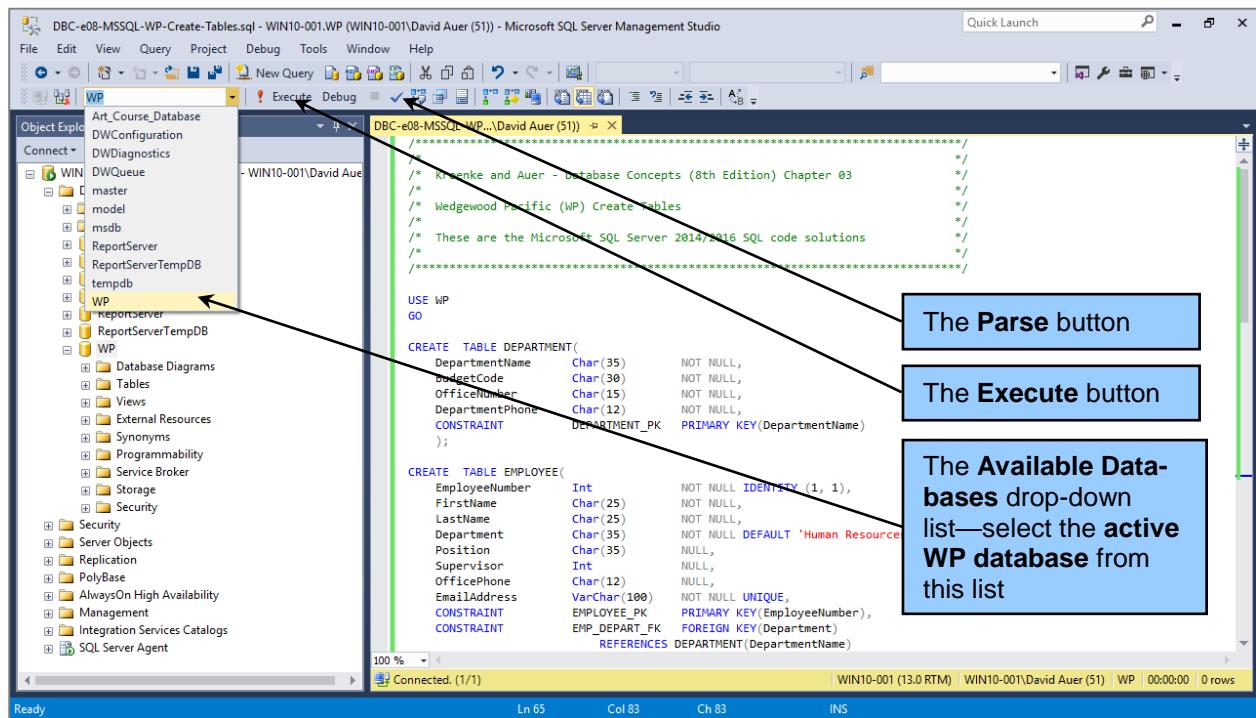


Figure A-21 — The Available Databases Drop-Down List

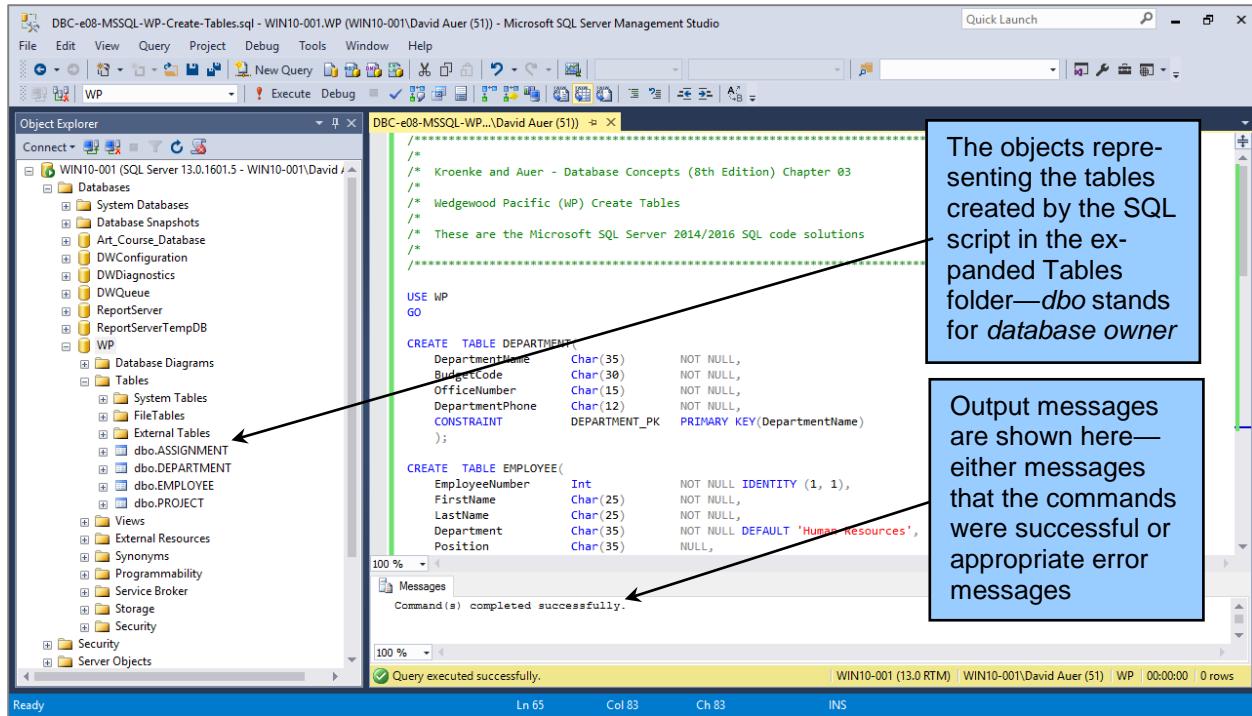


Figure A-22 — The SQL Script Results—The WP Tables

```

USE WP
GO

*****  DEPARTMENT DATA  *****

INSERT INTO DEPARTMENT VALUES(
    'Administration', 'BC-100-10', 'BLDG01-210', '360-285-8100');
INSERT INTO DEPARTMENT VALUES(
    'Legal', 'BC-200-10', 'BLDG01-220', '360-285-8200');
INSERT INTO DEPARTMENT VALUES(
    'Human Resources', 'BC-300-10', 'BLDG01-230', '360-285-8300');
INSERT INTO DEPARTMENT VALUES(
    'Finance', 'BC-400-10', 'BLDG01-110', '360-285-8400');
INSERT INTO DEPARTMENT VALUES(
    'Accounting', 'BC-500-10', 'BLDG01-120', '360-285-8405');
INSERT INTO DEPARTMENT VALUES(
    'Sales and Marketing', 'BC-600-10', 'BLDG01-250', '360-287-8500');
INSERT INTO DEPARTMENT VALUES(
    'InfoSystems', 'BC-700-10', 'BLDG02-210', '360-287-8600');
INSERT INTO DEPARTMENT VALUES(
    'Research and Development', 'BC-700-10', 'BLDG02-250', '360-287-8700');
INSERT INTO DEPARTMENT VALUES(
    'Production', 'BC-800-10', 'BLDG02-110', '360-287-8800');

```

Figure A-23 — The Figure 3-12 SQL Statements to Populate the WP Database Tables

```
***** EMPLOYEE DATA *****/
INSERT INTO EMPLOYEE (FirstName, LastName, Department, Position, OfficePhone,
EmailAddress) VALUES(
'Mary', 'Jacobs', 'Administration', 'CEO', '360-285-8110', 'Mary.Jacobs@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Rosalie', 'Jackson', 'Administration', 'Admin Assistant', 1,
'360-285-8120', 'Rosalie.Jackson@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Richard', 'Bandalone', 'Legal', 'Attorney', 1,
'360-285-8210', 'Richard.Bandalone@WP.com');
INSERT INTO EMPLOYEE VALUES(
'George', 'Smith', 'Human Resources', 'HR3', 1,
'360-285-8310', 'George.Smith@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Alan', 'Adams', 'Human Resources', 'HR1', 4,
'360-285-8320', 'Alan.Adams@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Ken', 'Evans', 'Finance', 'CFO', 1,
'360-285-8410', 'Ken.Evans@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Mary', 'Abernathy', 'Finance', 'FA3', 6,
'360-285-8420', 'Mary.Abernathy@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Tom', 'Caruthers', 'Accounting', 'FA2', 6,
'360-285-8430', 'Tom.Caruthers@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Heather', 'Jones', 'Accounting', 'FA2', 6,
'360-285-8440', 'Heather.Jones@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Ken', 'Numoto', 'Sales and Marketing', 'SM3', 1,
'360-287-8510', 'Ken.Numoto@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Linda', 'Granger', 'Sales and Marketing', 'SM2', 10,
'360-287-8520', 'Linda.Granger@WP.com');
INSERT INTO EMPLOYEE VALUES(
'James', 'Nestor', 'InfoSystems', 'CIO', 1,
'360-287-8610', 'James.Nestor@WP.com');
INSERT INTO EMPLOYEE(FirstName, LastName, Department, Position, Supervisor, EmailAddress)
VALUES(
'Rick', 'Brown', 'InfoSystems', 'IS2', 12, 'Rick.Brown@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Mike', 'Nguyen', 'Research and Development', 'CTO', 1,
'360-287-8710', 'Mike.Nguyen@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Jason', 'Sleeman', 'Research and Development', 'RD3', 14,
'360-287-8720', 'Jason.Sleeman@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Mary', 'Smith', 'Production', 'OPS3', 1,
'360-287-8810', 'Mary.Smith@WP.com');
INSERT INTO EMPLOYEE VALUES(
'Tom', 'Jackson', 'Production', 'OPS2', 14,
'360-287-8820', 'Tom.Jackson@WP.com');
INSERT INTO EMPLOYEE VALUES(
'George', 'Jones', 'Production', 'OPS2', 15,
```

Figure A-23 — The Figure 3-12 SQL Statements to Populate the WP Database Tables (continued)

```
'360-287-8830', 'George.Jones@WP.com');
INSERT INTO EMPLOYEE(FirstName, LastName, Department, Position, Supervisor, EmailAddress)
VALUES(
'Julia', 'Hayakawa', 'Production', 'OPS1', 15, 'Julia.Hayakawa@WP.com');
INSERT INTO EMPLOYEE(FirstName, LastName, Department, Position, Supervisor, EmailAddress)
VALUES(
'Sam', 'Stewart', 'Production', 'OPS1', 15, 'Sam.Stewart@WP.com');

***** PROJECT DATA *****/
INSERT INTO PROJECT VALUES(
'2017 Q3 Production Plan', 'Production', 100.00, '10-MAY-17', '15-JUN-17');
INSERT INTO PROJECT VALUES(
'2017 Q3 Marketing Plan', 'Sales and Marketing', 135.00, '10-MAY-17', '15-JUN-17');
INSERT INTO PROJECT VALUES(
'2017 Q3 Portfolio Analysis', 'Finance', 120.00, '05-JUL-17', '25-JUL-17' );
INSERT INTO PROJECT VALUES(
'2017 Q3 Tax Preparation', 'Accounting', 145.00, '10-AUG-17', '15-OCT-17');
INSERT INTO PROJECT VALUES(
'2017 Q4 Production Plan', 'Production', 100.00, '10-AUG-17', '15-SEP-17');
INSERT INTO PROJECT VALUES(
'2017 Q4 Marketing Plan', 'Sales and Marketing', 135.00, '10-AUG-17', '15-SEP-17');
INSERT INTO PROJECT (ProjectName, Department, MaxHours, StartDate)
VALUES(
'2017 Q4 Portfolio Analysis', 'Finance', 140.00, '05-OCT-17');

***** ASSIGNMENT DATA *****/
INSERT INTO ASSIGNMENT VALUES(1000, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1000, 6, 50.0);
INSERT INTO ASSIGNMENT VALUES(1000, 10, 50.0);
INSERT INTO ASSIGNMENT VALUES(1000, 16, 75.0);
INSERT INTO ASSIGNMENT VALUES(1000, 17, 75.0);
INSERT INTO ASSIGNMENT VALUES(1100, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1100, 6, 75.0);
INSERT INTO ASSIGNMENT VALUES(1100, 10, 55.0);
INSERT INTO ASSIGNMENT VALUES(1100, 11, 55.0);
INSERT INTO ASSIGNMENT VALUES(1200, 3, 20.0);
INSERT INTO ASSIGNMENT VALUES(1200, 6, 40.0);
INSERT INTO ASSIGNMENT VALUES(1200, 7, 45.0);
INSERT INTO ASSIGNMENT VALUES(1200, 8, 45.0);
INSERT INTO ASSIGNMENT VALUES(1300, 3, 25.0);
INSERT INTO ASSIGNMENT VALUES(1300, 6, 40.0);
INSERT INTO ASSIGNMENT VALUES(1300, 8, 50.0);
INSERT INTO ASSIGNMENT VALUES(1300, 9, 50.0);
INSERT INTO ASSIGNMENT VALUES(1400, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1400, 6, 50.0);
INSERT INTO ASSIGNMENT VALUES(1400, 10, 50.0);
INSERT INTO ASSIGNMENT VALUES(1400, 16, 75.0);
INSERT INTO ASSIGNMENT VALUES(1400, 17, 75.0);
INSERT INTO ASSIGNMENT VALUES(1500, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1500, 6, 75.0);
```

Figure A-23 — The Figure 3-12 SQL Statements to Populate the WP Database Tables (continued)

```
INSERT INTO ASSIGNMENT VALUES(1500, 10, 55.0);
INSERT INTO ASSIGNMENT VALUES(1500, 11, 55.0);
INSERT INTO ASSIGNMENT VALUES(1600, 3, 20.0);
INSERT INTO ASSIGNMENT VALUES(1600, 6, 40.0);
INSERT INTO ASSIGNMENT VALUES(1600, 7, 45.0);
INSERT INTO ASSIGNMENT VALUES(1600, 8, 45.0);
```

Figure A-23 — The Figure 3-12 SQL Statements to Populate the WP Database Tables (continued)

How Do I Work with SQL Queries in Microsoft SQL Server?

Now that we've created and populated the WP 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. To run a query, be sure the Databases folder is expanded in the Object Browser so that you can see the database names. Click the **New Query** button in the Standard toolbar, as shown in Figure A-18, to display a new tabbed SQL window along with the SQL Editor toolbar. Then specify the database you want to query by selecting the database name in the Available Databases drop-down list to select it.

By now you should be familiar with using the SQL query tabbed document window and running SQL scripts.

To create and run an SQL query:

1. In the new SQL query window, type the text of the SQL query you want to run.
2. Click the parse button to test it. Correct any errors.
3. Click the Execute button in the SQL Editor toolbar.
4. If you want to save the query, you can save it just as you would any other *.sql script.

The query results appear in a tabbed Results window below the query window in a spreadsheet-style display, as shown in Figure A-24. You can adjust the size of the query window and the Results window, and 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 also have multiple queries open at the same time—clicking the New Query button again opens another tabbed query window.

How Do I Install the Microsoft SQL Server 2016 ODBC Client?

For SQL Server 2016, you do not have to take any extra steps to install ODBC support. The ODBC Driver 13 for SQL Server (and the older SQL Server Native Client 11.0) is automatically installed and available for use, as shown in Figure 7-9.

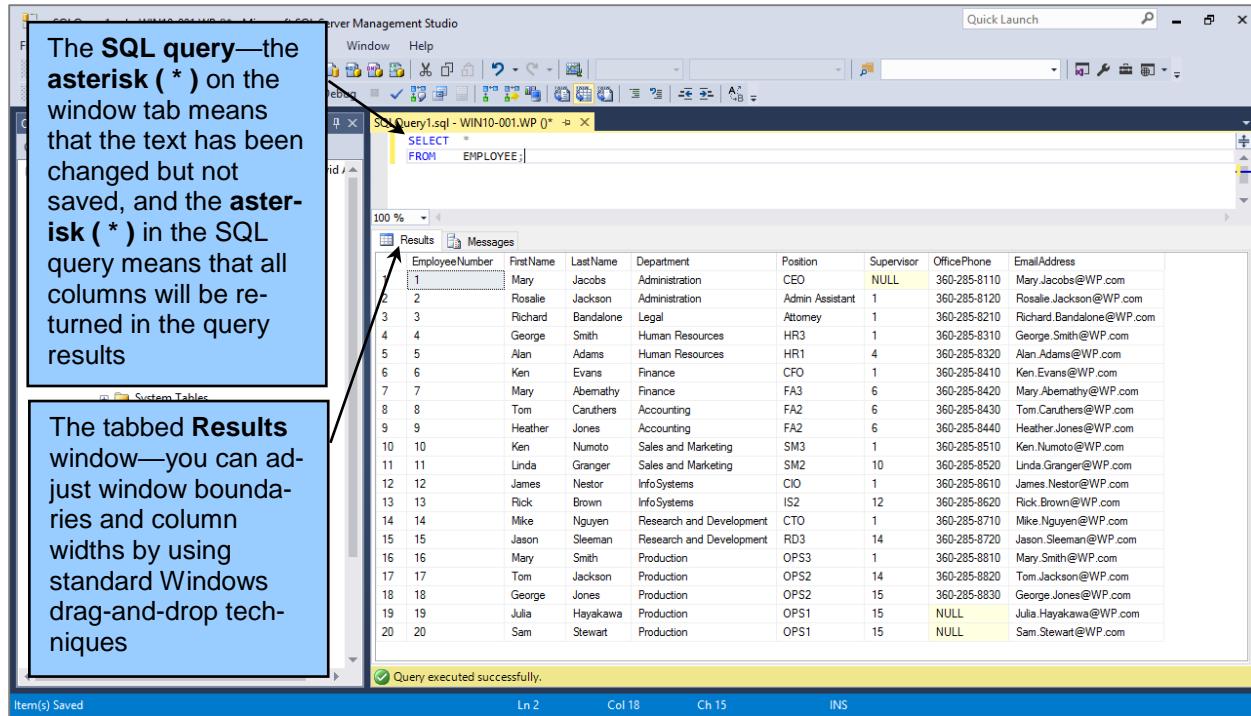


Figure A-24 — The SQL Query Results

By The Way

Before proceeding with the rest of the material in this Appendix, we recommend that you work through and understand the SQL topics covered in Chapter 3, "Structured Query Language," and Appendix E, "Advanced SQL."

How Do I Import Microsoft Excel Data into a Microsoft SQL Server Database Table?

When developing a database to support an application, it is very common to find that some (if not all) of the data needed in the database exists as data in user **worksheets** (also called **spreadsheets**). A typical example of this is a Microsoft Excel 2016 worksheet that a user has been maintaining that must now be converted to data stored in the database.

If we are really lucky, the worksheet will already be organized like a database table, with appropriate column labels and unique data in each row. And if we are *really, really lucky*, there will be one or more columns that can be used as the primary key in the new database table. In that case, we can easily import the data into the database. More likely, we will have to modify the worksheet, and then organize and clean up the data in it before we can import the data. In essence, we are following a procedure that we will encounter again in Chapter 8 in our discussion of data warehouses known as **extract, transform, and load (ETL)**.

As an example, let's consider the problem of computers owned by WP. WP needs to track these computers (asset inventory) and who they are currently and have previously been assigned to for use. The properly designed tables (COMPUTER and COMPUTER_ASSIGNMENT) to handle this problem are shown in the Chapter 3 Access Workbench Exercises as Figures 3-32 and 3-34. The data for the tables is shown in Figures 3-33 and 3-35.

Unfortunately, that is not the way we will probably encounter the data. More likely we'll find it stored in a worksheet such as the Microsoft Excel 2016 worksheet shown in Figure A-25. This worksheet breaks our basic rule of one theme per table, and combines computer inventory and computer assignment data into the same worksheet. Worse, the computer assignments are handled by using multiple assignment and date columns.

This is an example of what is called the **multivalue, multicolumn problem**, which occurs when multiple columns are used in a spreadsheet or database table to record repetitions of the same data. A good example is EMPLOYEE phone number data, where we might find columns for HomePhone, CellPhone, and BusinessPhone. This may seem reasonable until we have to add yet *another* phone number, perhaps DepartmentPhone or SpousesPhone.

What we are dealing with here is a **multivalued dependency** (as discussed in Chapter 2), where the determinant determines multiple values instead of just one:

EmployeeID →→ PhoneNumber

A detailed solution to this problem is beyond the scope of this book, but the basic answer to use 4NF and put this dependency into its own table.⁴

In our current situation, it is obvious that we must *extract* the data we need from the worksheet for two database tables—COMPUTER and COMPUTER_ASSIGNMENT—*transform* each set of data into a correctly structured and formatted worksheet, and then *load* (import) the data from the worksheets into the database.

We can do this by:

- Creating two new worksheets named COMPUTER and COMPUTER_ASSIGNMENT and copying the data into them, then
- Modify the structure and data in each worksheet so that it is correct for importing into the database, then
- Import the data from each worksheet into the database.

After the data is imported into two database tables, we will then have to use SQL ALTER TABLE statements to create primary key, foreign keys, and any other needed constraints—while we will use the SQL ALTER TABLE statement as needed here, a full discussion of the SQL ALTER TABLE statement can be found in Appendix E, “Advanced SQL.”

⁴ For more information about multivalued dependencies and the multivalue, multicolumn problem, see David Kroenke and David Auer, *Database Processing: Fundamentals, Design, and Implementation*, 14th edition (Upper Saddle River, NJ: Pearson Higher Education, 2016).

Figure A-25 – The WP Computer Assignments in a Microsoft Excel 2016 Worksheet

Preparing the Microsoft Excel Data for Import into a Database Table

Figure A-26 shows the WP COMPUTER worksheet after it has been cleaned up. All extraneous rows and columns have been deleted, and only the computer data (with appropriate column headers) remains. This worksheet now looks like a database table, which is a good indication that the data import should work properly.

Figure A-27 shows the COMPUTER_ASSIGNMENT worksheet after our first attempt at restructuring it. There are still some obvious problems here. First of all, in the WP database we identify employees by their *EmployeeNumber*, not by their name. Second, we still have multiple Assigned To and Date columns. Therefore, we need to (1) substitute *EmployeeNumber* for *Assigned To* and (2) combine the Assigned to and Date columns. We can determine EmployeeNumber (a surrogate key) by using an SQL query in the WP database:

```
/* *** SQL-Query-AppA-01 *** */
SELECT * FROM EMPLOYEE;
```

This query gives us:

	EmployeeNumber	FirstName	LastName	Department	Position	Supervisor	OfficePhone	EmailAddress
1	1	Mary	Jacobs	Administration	CEO	NULL	360-285-8110	Mary.Jacobs@WP.com
2	2	Rosalie	Jackson	Administration	Admin Assistant	1	360-285-8120	Rosalie.Jackson@WP.com
3	3	Richard	Bandalone	Legal	Attorney	1	360-285-8210	Richard.Bandalone@WP.com
4	4	George	Smith	Human Resources	HR3	1	360-285-8310	George.Smith@WP.com
5	5	Alan	Adams	Human Resources	HR1	4	360-285-8320	Alan.Adams@WP.com
6	6	Ken	Evans	Finance	CFO	1	360-285-8410	Ken.Evans@WP.com
7	7	Mary	Abermathy	Finance	FA3	6	360-285-8420	Mary.Abermathy@WP.com
8	8	Tom	Caruthers	Accounting	FA2	6	360-285-8430	Tom.Caruthers@WP.com
9	9	Heather	Jones	Accounting	FA2	6	360-285-8440	Heather.Jones@WP.com
10	10	Ken	Numoto	Sales and Marketing	SM3	1	360-285-8510	Ken.Numoto@WP.com
11	11	Linda	Granger	Sales and Marketing	SM2	10	360-285-8520	Linda.Granger@WP.com
12	12	James	Nestor	InfoSystems	CIO	1	360-285-8610	James.Nestor@WP.com
13	13	Rick	Brown	InfoSystems	IS2	12	360-285-8620	Rick.Brown@WP.com
14	14	Mike	Nguyen	Research and Development	CTO	1	360-285-8710	Mike.Nguyen@WP.com
15	15	Jason	Sleeman	Research and Development	RD3	14	360-285-8720	Jason.Sleeman@WP.com
16	16	Mary	Smith	Production	OPS3	1	360-285-8810	Mary.Smith@WP.com
17	17	Tom	Jackson	Production	OPS2	14	360-285-8820	Tom.Jackson@WP.com
18	18	George	Jones	Production	OPS2	15	360-285-8830	George.Jones@WP.com
19	19	Julia	Hayakawa	Production	OPS1	15	NULL	Julia.Hayakawa@WP.com
20	20	Sam	Stewart	Production	OPS1	15	NULL	Sam.Stewart@WP.com

Using this data, we can rework the COMPUTER_ASSIGNMENT worksheet as shown in Figure A-28, which also includes a renamed *Date* column which is now *DateAssigned*. Admittedly this is a small example, and given a larger data set a different strategy would be needed. For our purposes here, however, this method will work.

We will now look at how to import a table into Microsoft SQL Server 2016. The COMPUTER table column characteristics as stated in Figure 3-32 are shown below in Figure A-29. Note that we will need to CHECK constraints on this table and that neither of these can be done in the data import—we will have to use SQL ALTER TABLE statements to implement these constraints after the table is created and the data is imported.

Screenshot of Microsoft Excel showing the 'COMPUTER' worksheet from the 'DBC-e08-WP-Computer-Assignment-Worksheet.xlsx' file. The table contains data for various computer components.

SerialNumber	Make	Model	ProcessorType	ProcessorSpeed	MainMemory	DiskSize
9871234	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871235	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871236	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871237	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871238	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871239	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871240	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871241	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871242	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
9871243	HP	ProDesk 600 G1	Intel i5-4690	3.50	16.0 GBbytes	1.0 TBytes
6541001	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541002	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541003	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541004	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541005	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541006	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541007	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541008	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541009	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes
6541010	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GBbytes	2.0 TBytes

Figure A-26 — The WP COMPUTER Worksheet

Screenshot of Microsoft Excel showing the 'COMPUTER_ASSIGNMENT' worksheet from the 'DBC-e08-WP-Computer-Assignment-Worksheet.xlsx' file. The table contains data for computer assignments.

SerialNumber	Assigned To	Date	Assigned To	Date
9871234	James Nestor	15-Sep-17	Mary Jacobs	21-Oct-17
9871235	Rick Brown	15-Sep-17	Rosalie Jackson	21-Oct-17
9871236	Mike Nguyen	15-Sep-17	Richard Bandalone	21-Oct-17
9871237	Jason Sleeman	15-Sep-17	George Smith	21-Oct-17
9871238	Ken Evans	15-Sep-17	Alan Adams	21-Oct-17
9871239	Mary Abernathy	15-Sep-17	Ken Numoto	21-Oct-17
9871240	Tom Caruthers	15-Sep-17	Linda Granger	21-Oct-17
9871241	Heather Jones	15-Sep-17	George Jones	21-Oct-17
9871242	Mary Smith	15-Sep-17	Julia Hayakawa	21-Oct-17
9871243	Tom Jackson	15-Sep-17	Sam Stewart	21-Oct-17
6541001	James Nestor	21-Oct-17		
6541002	Rick Brown	21-Oct-17		
6541003	Mike Nguyen	21-Oct-17		
6541004	Jason Sleeman	21-Oct-17		
6541005	Ken Evans	21-Oct-17		
6541006	Mary Abernathy	21-Oct-17		
6541007	Tom Caruthers	21-Oct-17		
6541008	Heather Jones	21-Oct-17		
6541009	Mary Smith	21-Oct-17		
6541010	Tom Jackson	21-Oct-17		

Figure A-27 — The WP COMPUTER_ASSIGNMENT Worksheet — First Attempt

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	SerialNumber	EmployeeNumber	DateAssigned													
2	9871234	12	15-Sep-17													
3	9871235	13	15-Sep-17													
4	9871236	14	15-Sep-17													
5	9871237	15	15-Sep-17													
6	9871238	6	15-Sep-17													
7	9871239	7	15-Sep-17													
8	9871240	8	15-Sep-17													
9	9871241	9	15-Sep-17													
10	9871242	16	15-Sep-17													
11	9871243	17	15-Sep-17													
12	6541001	12	21-Oct-17													
13	6541002	13	21-Oct-17													
14	6541003	14	21-Oct-17													
15	6541004	15	21-Oct-17													
16	6541005	6	21-Oct-17													
17	6541006	7	21-Oct-17													
18	6541007	8	21-Oct-17													
19	6541008	9	21-Oct-17													
20	6541009	16	21-Oct-17													
21	6541010	17	21-Oct-17													
22	9871234	1	21-Oct-17													
23	9871235	2	21-Oct-17													
24	9871236	3	21-Oct-17													
25	9871237	4	21-Oct-17													
26	9871238	5	21-Oct-17													
27	9871239	10	21-Oct-17													
28	9871240	11	21-Oct-17													
29	9871241	18	21-Oct-17													
30	9871242	19	21-Oct-17													
31	9871243	20	21-Oct-17													
32																
33																

Figure A-28 — The WP COMPUTER_ASSIGNMENT Worksheet — Second Attempt

Database Column Characteristics for the WP COMPUTER Table

Column Name	Type	Key	Required	Remarks
SerialNumber	Number	Primary Key	Yes	Long Integer
Make	Short Text (12)	No	Yes	Must be “Dell” or “Gateway” or “HP” or “Other”
Model	Short Text (24)	No	Yes	
ProcessorType	Short Text (24)	No	No	
ProcessorSpeed	Number	No	Yes	Double [3,2], Between 1.0 and 4.0
MainMemory	Short Text (15)	No	Yes	
DiskSize	Short Text (15)	No	Yes	

Figure A-29 — Database Column Characteristics for the WP COMPUTER Table

Importing the Microsoft Excel Data into a Microsoft SQL Server 2016 Database Table

Because Microsoft creates both Microsoft Excel 2016 and Microsoft SQL Server 2016, we would expect that importing data from Microsoft Excel into SQL Server would be simple and problem free. Unfortunately, in our experience, the **SQL Server Import and Export Wizard**, which is the tool we use for data import, has some glitches.

First, although the current SQL Server 2016 version of the SQL Server Import and Export Wizard appears programmed to work with Microsoft Excel 2016 workbooks (and Microsoft Excel 2013 workbooks), a driver is missing, and if we try to use the default installation, we will get an error message. Since we are using Microsoft Excel 2016, we have to download and install the **Microsoft Access Database Engine 2010 Redistributable** from <https://www.microsoft.com/en-us/download/details.aspx?id=13255>. There are both 32-bit and 64-bit versions—install the 32-bit version if you are running a 32-bit version of Microsoft Office or the 64-bit version if you are running a 64-bit version of Office. If you don't install this software, you will get an error message during the Wizard, and it will not complete its tasks.⁵

Second, the Wizard does not handle data types or NULL/NOT NULL constraints smoothly. We cannot change the Wizard-detected data types or NULL/NOT NULL settings into the data types we want in our database—if we try, the Wizard generates an error message and will not complete its tasks.

Third, the Wizard does not allow a primary key to be set on the imported table and also imports a set of blank rows (all NULL values) in addition to the actual data (this is only possible because no primary key has been set.) Our solution is to:

- Use the SQL Server Import and Export Wizard to import the data in a temporary table as created by the Wizard, then
- Use an SQL CREATE TABLE statement to create the actual table we want in the database, then
- Use an SQL INSERT statement to copy the data from the temporary table to the actual table, then
- Delete the temporary table from the database.

Note that in these steps will use a new variant of the SQL INSERT statement, a **bulk INSERT statement**. We use this form of the SQL INSERT statement when we want to copy a lot of data from one table to another, and copying from a temporary table to a final table is a great place to use this statement. In this case, given the name of the temporary table will be *COMPUTER\$*, the SQL statement will be:

```
/* *** SQL-INSERT-AppA-01 *** */
INSERT INTO dbo.COMPUTER
    (SerialNumber, Make, Model, ProcessorType,
     ProcessorSpeed, MainMemory, DiskSize)
SELECT    SerialNumber, Make, Model, ProcessorType,
          ProcessorSpeed, MainMemory, DiskSize
FROM      COMPUTER$
WHERE    SerialNumber IS NOT NULL;
```

⁵ This statement is true on Windows 10 running Microsoft Office 2016 and Microsoft SQL Server 2016 Developer Edition with all updates and patches installed as of August 8, 2016. Hopefully Microsoft will update the SQL Server Import and Export Wizard and its supporting software in the near future. Note that there is also a **Microsoft Access 2013 Runtime** available at <https://www.microsoft.com/en-us/download/details.aspx?id=39358>.

Note the use of the imbedded SQL SELECT statement where we would expect to find a VALUES clause.

Here are the actual steps:

1. In the Microsoft SQL Server Management Studio, expand the **WP** database.
2. Right-click on the **WP** database object to display a shortcut menu, and in the shortcut menu click on the **Tasks** command to display the Tasks menu, as shown in Figure A-30.
3. In the Task menu, click the **Import Data** command shown in Figure A-30 to launch the SQL Server Import and Export Wizard as shown in Figure A-31.
4. On the *Welcome to SQL Server Import and Export Wizard* page shown in Figure A-31, click the **Next** button to display the Choose a Data Source page as shown in Figure A-32.
5. On the *Choose a Data Source* page shown in Figure A-32, select **Microsoft Excel** as the data source.
6. On the *Choose a Data Source* page shown in Figure A-32, select browse to the location of the Microsoft Excel file, select the 2007-2010 version of **Microsoft Excel** listed in Excel version drop-down list (there is a glitch if Excel 2013 or Excel 2016 is used), and make sure the check box for First row has column names is checked, as shown in Figure A-32.
7. Click the **Next** button to display the *Choose a Destination* page as shown in Figure A-33, and select **SQL Server Native Client 11** as the destination. The WP database values are automatically supplied, and there is nothing to change.
8. Click the **Next** button to display the *Specify Table Copy or Query* page as shown in Figure A-34. Select the **Copy data from one of more tables or views** radio button.

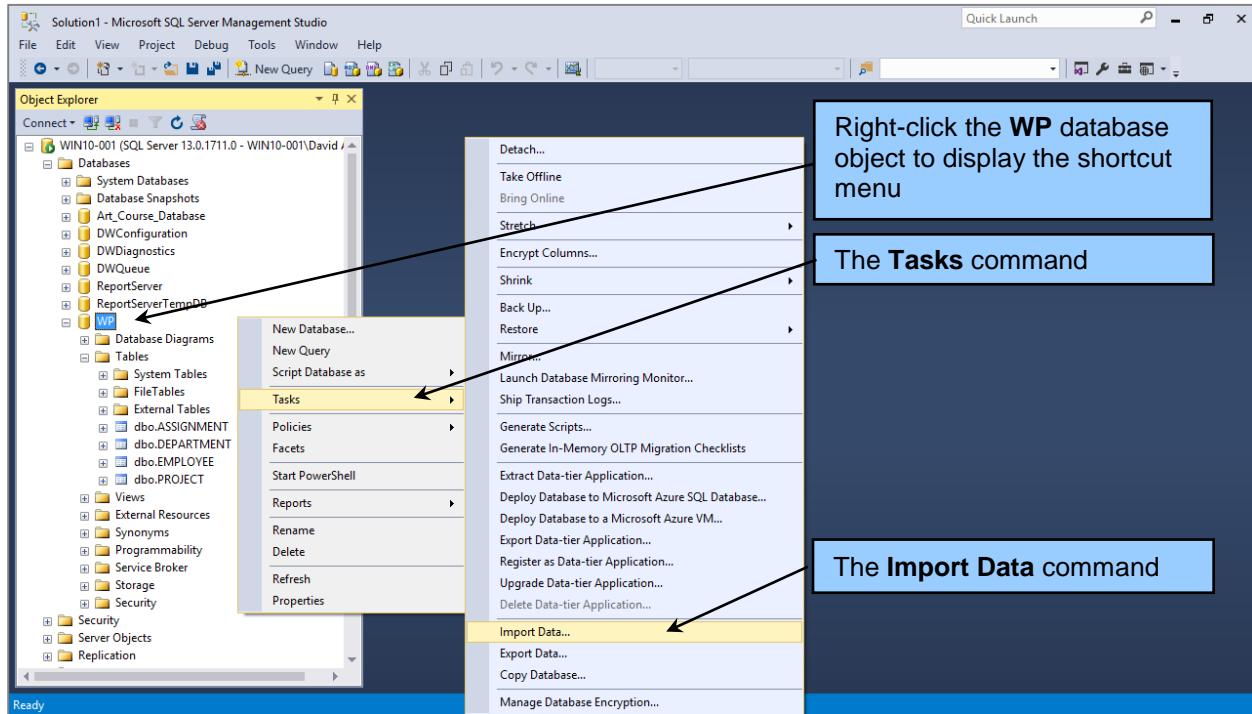


Figure A-30 — Launching the Microsoft SQL Server Import and Export Wizard

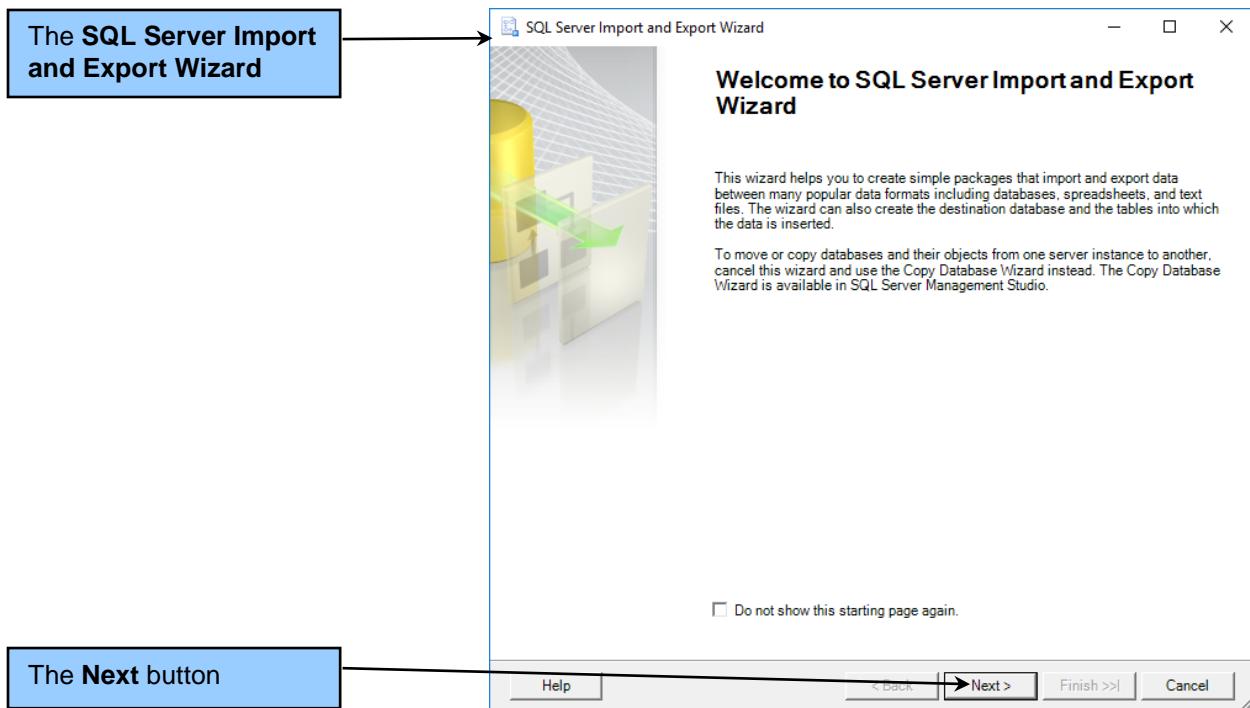


Figure A-31 — The Microsoft SQL Server Import and Export Wizard

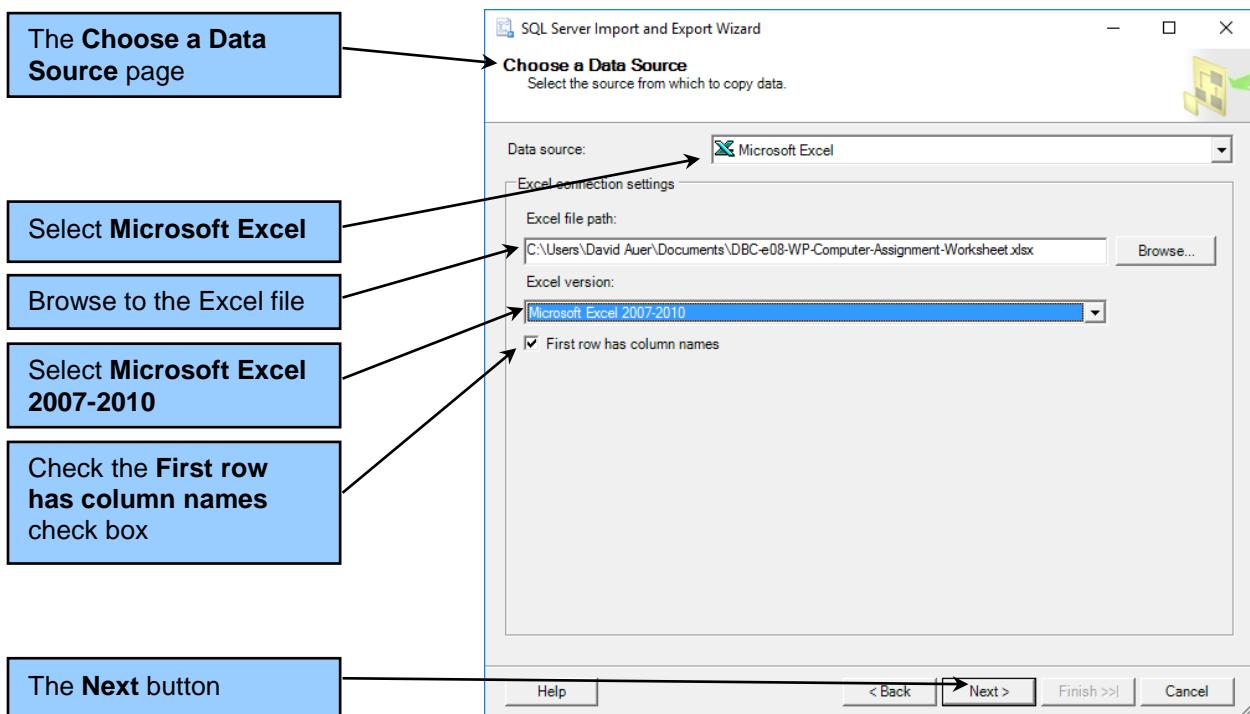


Figure A-32 — The Choose a Data Source Page

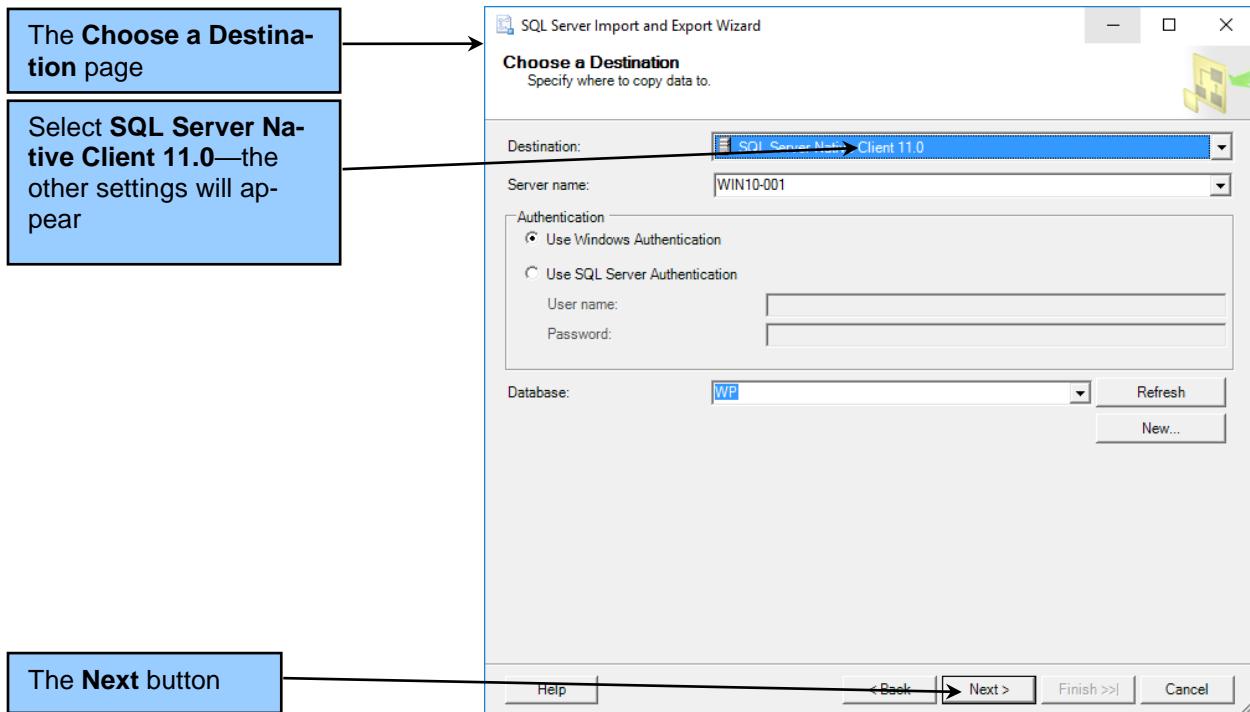


Figure A-33 — The Choose a Destination Page

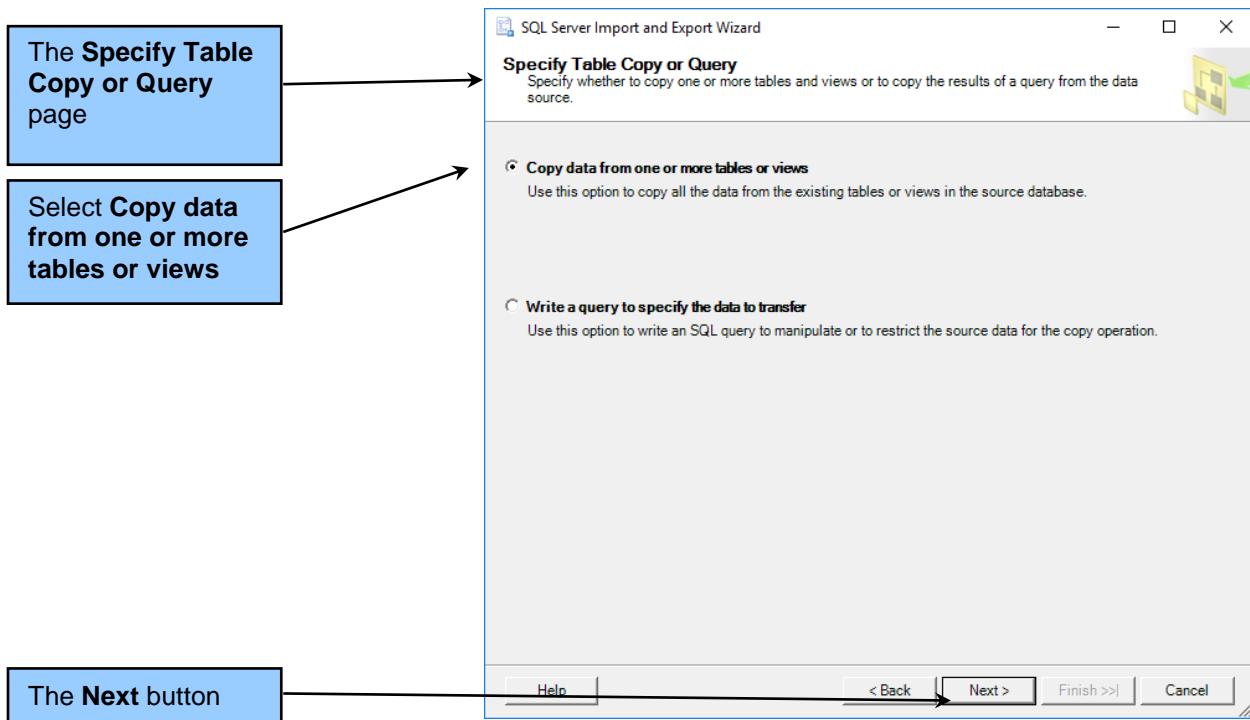


Figure A-34 — The Specify Table Copy or Query Page

9. Click the **Next** button to display the *Select Source Tables and Views* page as shown in Figure A-35, and check the ‘**COMPUTER\$**’ check box in the *Source* column. The table name *[dbo].[COMPUTER\$]* is generated and displayed in the *Destination* column. This is the name we will use for the temporary table in the WP database.
10. Click the **Edit Mappings** button to display the *Column Mappings* dialog box shown in Figure A-36. This dialog box shows the column names, data types, and NULL/NOT NULL settings that will be used to create the COMPUTER\$ table during the import.
 - **Note:** We should be able to edit these values, but if we do we are likely to generate errors during the import process. Therefore, we leave them alone and leave the temporary COMPUTER\$ table as created by the Wizard.
 - **Note:** You may want to try some other imports where you do edit these values in order to understand what you can and cannot successfully edit. When in doubt, leave it alone!
11. Click the **OK** button to return to the *Select Source Tables and Views* page, and then click the **Next** button.
12. The *Save and Run Package* page is displayed as shown in Figure A-37. Click the **Next** button to display the *Complete the Wizard* page as shown in Figure A-38 and then click the **Finish** button.
13. The SQL Server Import and Export Wizard runs the actual import, and then displays the *The execution was successful* page as shown in Figure A-39. Note that there are no errors in the process. Click the **Close** button to close the Wizard.

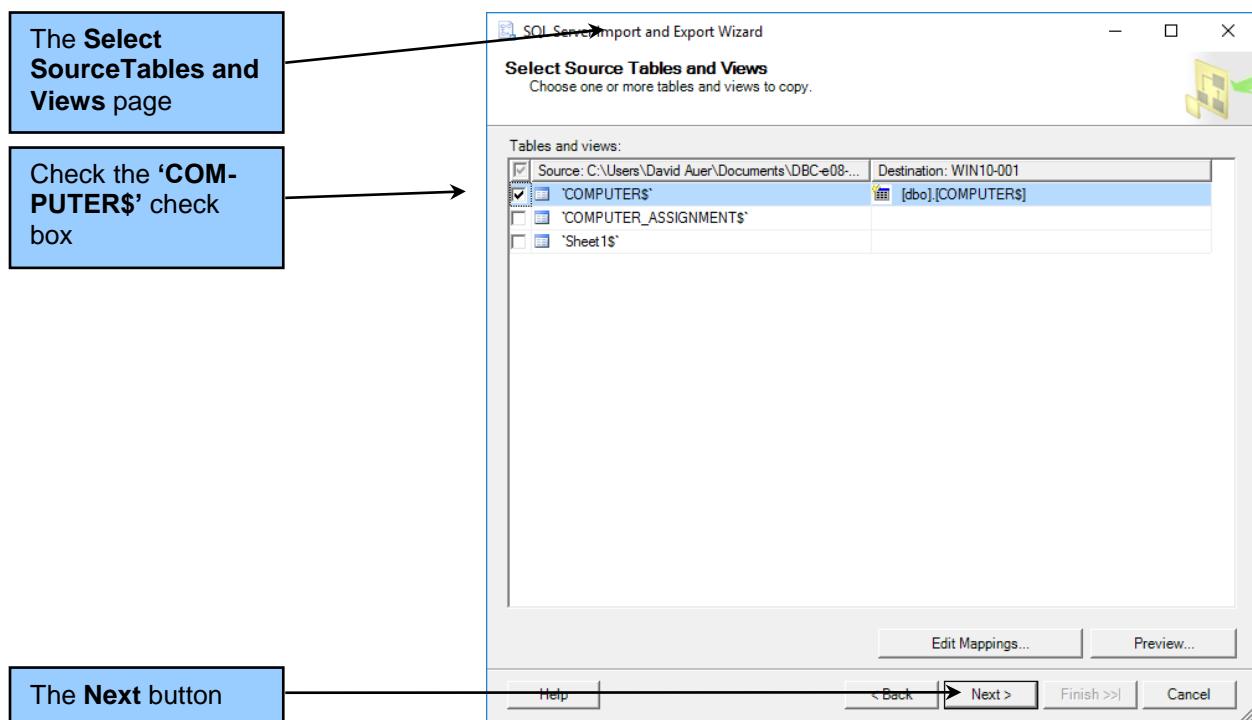


Figure A-35 — The Select Source Tables and Views Page

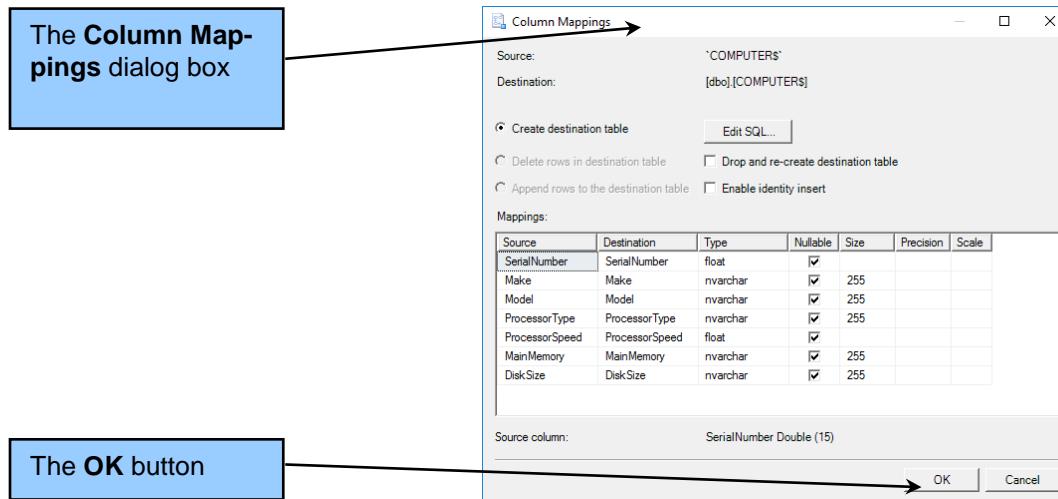


Figure A-36 — The Column Mappings Dialog Box

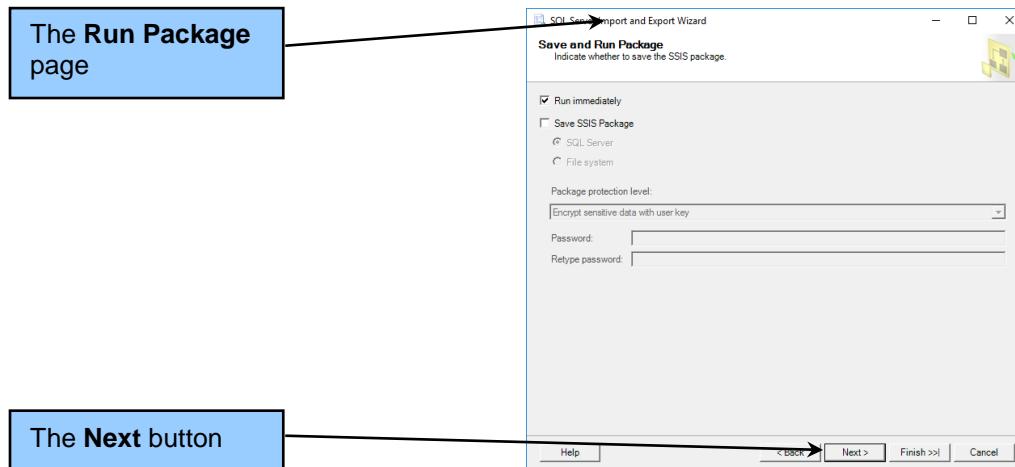


Figure A-37 — The Run Package Page

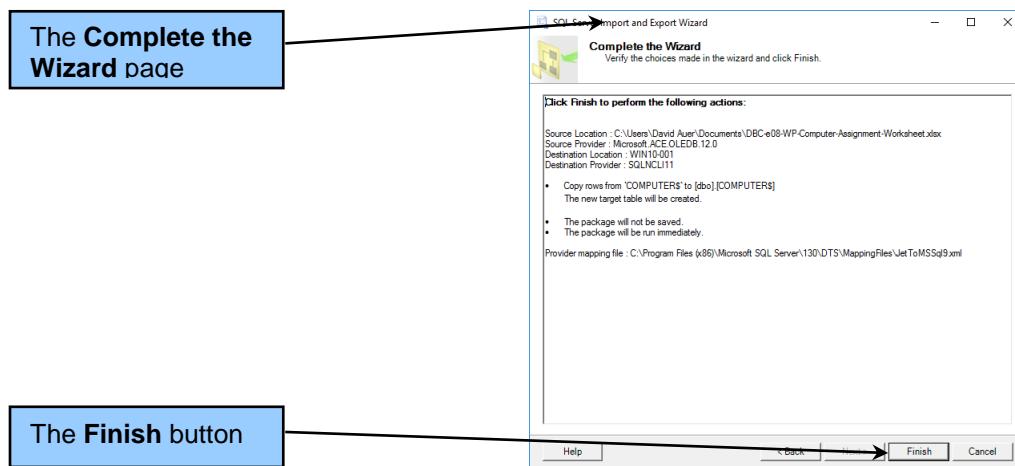


Figure A-38 — The Complete the Wizard Page

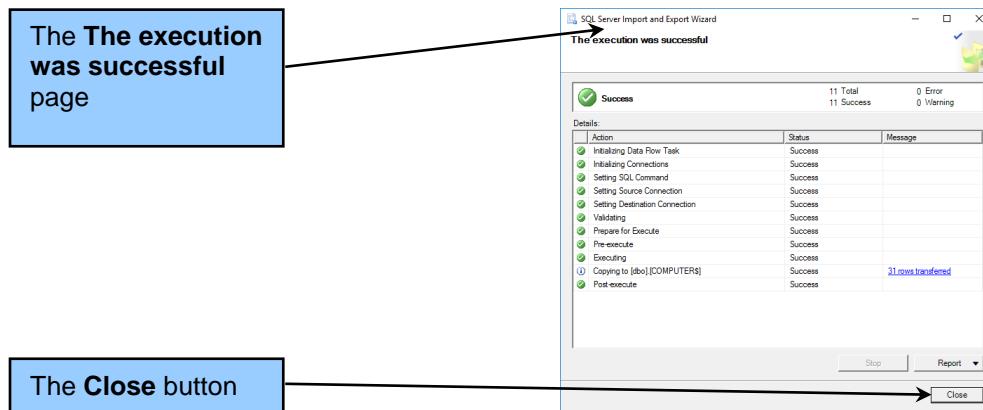


Figure A-39 — The Execution Was Successful Page

14. In SQL Server Management Studio, refresh the **WP** database. In Object Explorer, expand the **WP** database, then expand the **Tables** object, then expand the **dbo.COMPUTER\$** object, and finally expand the **Columns** object.
15. Open a New Query window, and run SQL-Query-AppA-02:

```
/* *** SQL-Query-AppA-02 *** */
SELECT * FROM COMPUTER$;
```

16. The results of SQL-Query-AppA-02 are shown in Figure A-40. Note that the SQL Server Import and Export Wizard inserted an additional 10 rows of blank data. This was only possible because no primary key was set and SerialNumber was allowed to be NULL during the import process.
17. Now we have to create the final COMPUTER table in the WP database. In the Microsoft SQL Server Management Studio, write the SQL CREATE TABLE statement for the COMPUTER table based on the column characteristics in Figure A-29 (these are Microsoft Access 2016 specifications). Note that in this case we can use the necessary CHECK CONSTRAINT statements as part of the CREATE TABLE statement and will not need to add them later. This will be the SQL-CREATE-TABLE-AppA-01:

```
/* *** SQL-CREATE-TABLE-AppA-01 *** */
CREATE TABLE COMPUTER (
    SerialNumber Int NOT NULL,
    Make Char(12) NOT NULL,
    Model Char(24) NOT NULL,
    ProcessorType Char(24) NULL,
    ProcessorSpeed Numeric(3,2) NOT NULL,
    MainMemory Char(15) NOT NULL,
    DiskSize Char(15) NOT NULL,
    CONSTRAINT COMPUTER_PK PRIMARY KEY(SerialNumber),
    CONSTRAINT MAKE_CHECK CHECK
        (Make IN ('Dell', 'Gateway', 'HP', 'Other')),
    CONSTRAINT SPEED_CHECK CHECK
        (ProcessorSpeed BETWEEN 1.0 AND 4.0)
);
```

18. Run the SQL-CREATE-TABLE-AppA-01 statement. The results are shown in Figure A-41.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, a database named 'WP' is selected. In the center, a query window titled 'SQLQuery1.sql - WIN10-001.WP 0*' displays the following SQL code:

```
SELECT *
FROM COMPUTER$;
```

The results grid shows a list of computer components with the following columns: SerialNumber, Make, Model, ProcessorType, ProcessorSpeed, MainMemory, and DiskSize. The data includes entries for various HP and Dell models with specific processor speeds and memory capacities.

SerialNumber	Make	Model	ProcessorType	ProcessorSpeed	MainMemory	DiskSize
9871234	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871235	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871236	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871237	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871238	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871239	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871240	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871241	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871242	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
9871243	HP	ProDesk 600 G1	Intel i5-4690	3.5	16.0 GB	1.0 TB
6541001	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541002	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541003	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541004	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541005	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541006	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541007	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541008	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541009	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
6541010	Dell	OptiPlex 7040	Intel i7-6700	3.4	32.0 GB	2.0 TB
NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query executed successfully.

Figure A-40 — The SQL-Query-AppA-02 Query and Results

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, a database named 'WP' is selected. In the center, a query window titled 'SQLQuery2.sql - WIN10-001.WP 0*' displays the following SQL code:

```
CREATE TABLE COMPUTER(
    SerialNumber Int NOT NULL,
    Make Char(12) NOT NULL,
    Model Char(24) NOT NULL,
    ProcessorType Char(24) NULL,
    ProcessorSpeed Numeric(3,2) NOT NULL,
    MainMemory Char(15) NOT NULL,
    DiskSize Char(15) NOT NULL,
    CONSTRAINT COMPUTER_PK PRIMARY KEY(SerialNumber),
    CONSTRAINT MAKE_CHECK CHECK (Make IN ('Dell', 'Gateway', 'HP', 'Other')),
    CONSTRAINT SPEED_CHECK CHECK (ProcessorSpeed BETWEEN 1.0 AND 4.0)
);
```

The results grid shows the message: "Command(s) completed successfully." Three callout boxes highlight specific elements: one points to the 'CREATE TABLE COMPUTER' statement, another points to the 'The COMPUTER table' entry in the Object Explorer, and a third points to the 'The COMPUTER\$ table' entry.

Figure A-41 — The SQL-CREATE-TABLE-AppA-01 Statement and Results

19. To copy the imported data from the temporary COMPUTER\$ table to the final COMPUTER table, use the SQL bulk INSERT statement SQL-INSERT-AppA-01:

```
/* *** SQL-INSERT-AppA-01 *** */
INSERT INTO dbo.COMPUTER
(SerialNumber, Make, Model, ProcessorType,
ProcessorSpeed, MainMemory, DiskSize)
SELECT SerialNumber, Make, Model, ProcessorType,
ProcessorSpeed, MainMemory, DiskSize
FROM COMPUTER$
WHERE SerialNumber IS NOT NULL;
```

20. After running the SQL-INSERT-AppA-01 statement, run SQL-Query-AppA-03:

```
/* *** SQL-Query-AppA-03 *** */
SELECT *
FROM COMPUTER;
```

21. The results for SQL-Query-AppA-03 are shown in Figure A-42. Note that we now have the correct 20 rows of data.

22. Drop the temporary COMPUTER\$ table (be sure you drop the right table!) using SQL-DROP-TABLE-AppA-01:

```
/* *** SQL-DROP-TABLE-AppA-01 *** */
DROP TABLE COMPUTER$;
```

Because we were able to put all needed constraints, including PRIMARY KEY and the CHECK constraints, into the SQL CREATE TABLE statement, the COMPUTER table does not require any modifications and is ready to use.

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer displays the database structure, including the 'Computer' table under the 'Tables' node. A callout box labeled 'The SQL bulk INSERT statement' points to the 'Computer' table node. Another callout box labeled 'Data in the COMPUTER table' points to the results grid on the right.

In the center, a query window titled 'SQLQuery3.sql - WIN10-001.WP [1]' contains the following SQL code:

```
/* *** SQL-INSERT-AppA-01 *** */
INSERT INTO dbo.COMPUTER
(SerialNumber, Make, Model, ProcessorType,
ProcessorSpeed, MainMemory, DiskSize)
SELECT SerialNumber, Make, Model, ProcessorType,
ProcessorSpeed, MainMemory, DiskSize
FROM COMPUTER$
WHERE SerialNumber IS NOT NULL;
```

Below the code, a 'Results' tab shows the output of the query, displaying 20 rows of data:

SerialNumber	Make	Model	ProcessorType	ProcessorSpeed	MainMemory	DiskSize
6541001	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541002	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541003	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541004	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541005	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541006	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541007	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541008	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541009	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
6541010	Dell	OptiPlex 7040	Intel i7-6700	3.40	32.0 GB/bytes	2.0 TB/tes
9871234	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871235	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871236	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871237	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871238	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871239	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871240	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871241	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871242	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes
9871243	HP	ProDesk 600 G1	Intel i5-4590	3.50	16.0 GB/bytes	1.0 TB/tes

At the bottom of the results grid, a message states 'Query executed successfully.'

Figure A-42 — The Final COMPUTER Table and Data

How Do I Create User Accounts in an SQL Server 2016 Database?

When you, as a database administrator, create a database in SQL Server 2016, you have full permissions within the DBMS to do whatever you need to with that database. However, in order for other people (or applications) to use that database, you must create appropriate user accounts with logins and appropriate permissions.

A full discussion of database administration and database security is provided in Chapter 6. Here, we will discuss the actual steps taken in an SQL Server 2016 database to create a user account and give it the appropriate permissions to read data from and write data to the WP database. We will then use the new user account as we demonstrate how to link a Microsoft Access 2016 database to an SQL Server 2016 database.

Creating a WP User Login with Permissions:

1. In the Microsoft SQL Server Management Studio, expand the **Security** folder so that the **Logins** folder and its contents are visible.
2. Right-click the **Logins** folder to display a shortcut menu, and click the **New Login** command, as shown in Figure A-43.
3. The **Login - New** dialog box is displayed with the **General** page selected, as shown in Figure A-44.

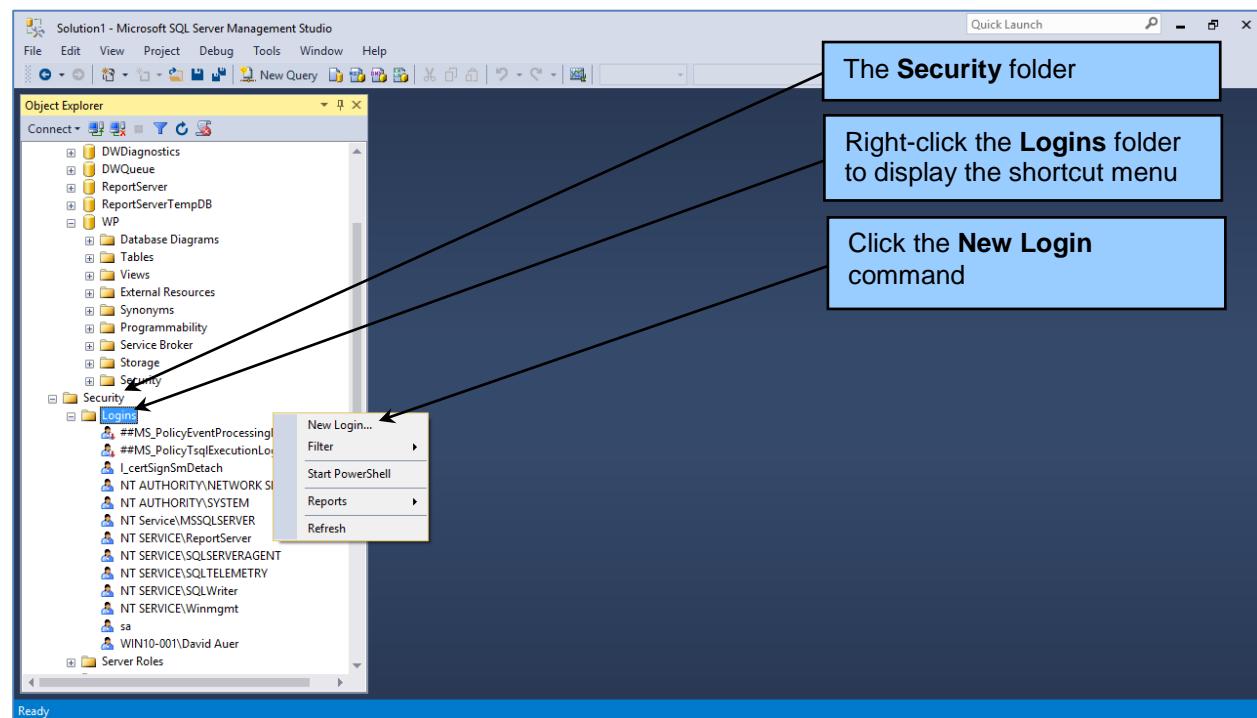


Figure A-43 — The New Login Command

4. In the **Login - New** dialog box General page, type in the login name **WP-User**, as shown in Figure A-44.
5. Click the **SQL Server authentication** radio button, as shown in Figure A-44.
6. Uncheck the **User must change password at next login** password setting, as shown in Figure A-44.
7. Uncheck the **Enforce password expiration** password setting, as shown in Figure A-44.
8. Type in the password **WP-User+password** in both the **Password** and **Confirm password** text boxes, as shown in Figure A-44.
9. Select **WP** as the **Default database** from the Default database drop-down list, as shown in Figure A-44.
10. Double check your settings, and then click the **User Mapping** button shown in Figure A-44 to display the User Mapping page of the Login – New dialog box, as shown in Figure A-45.
11. On the User Mapping page, scroll down the **Users mapped to this login** list until you can see the WP database settings, and click the check box in the Map column, as shown in Figure A-45.
12. In the **Database role memberships for:** WP list, leave already checked the **public** role selected and additionally check the check boxes for the **db_datareader** and **db_datawriter** database roles, as shown in Figure A-45. These permissions give WP-User the necessary rights to read data from the WP database tables and also to write new or revised data to the WP database tables.

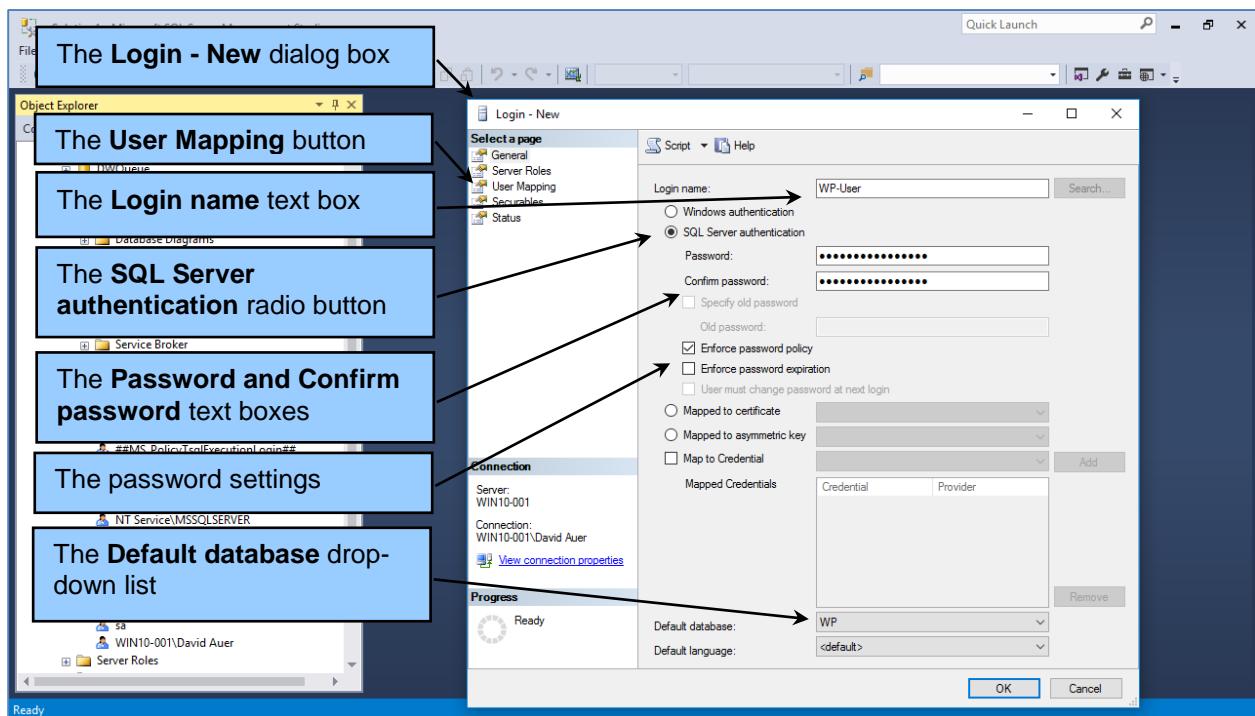


Figure A-44 — The Login – New Dialog Box General Page

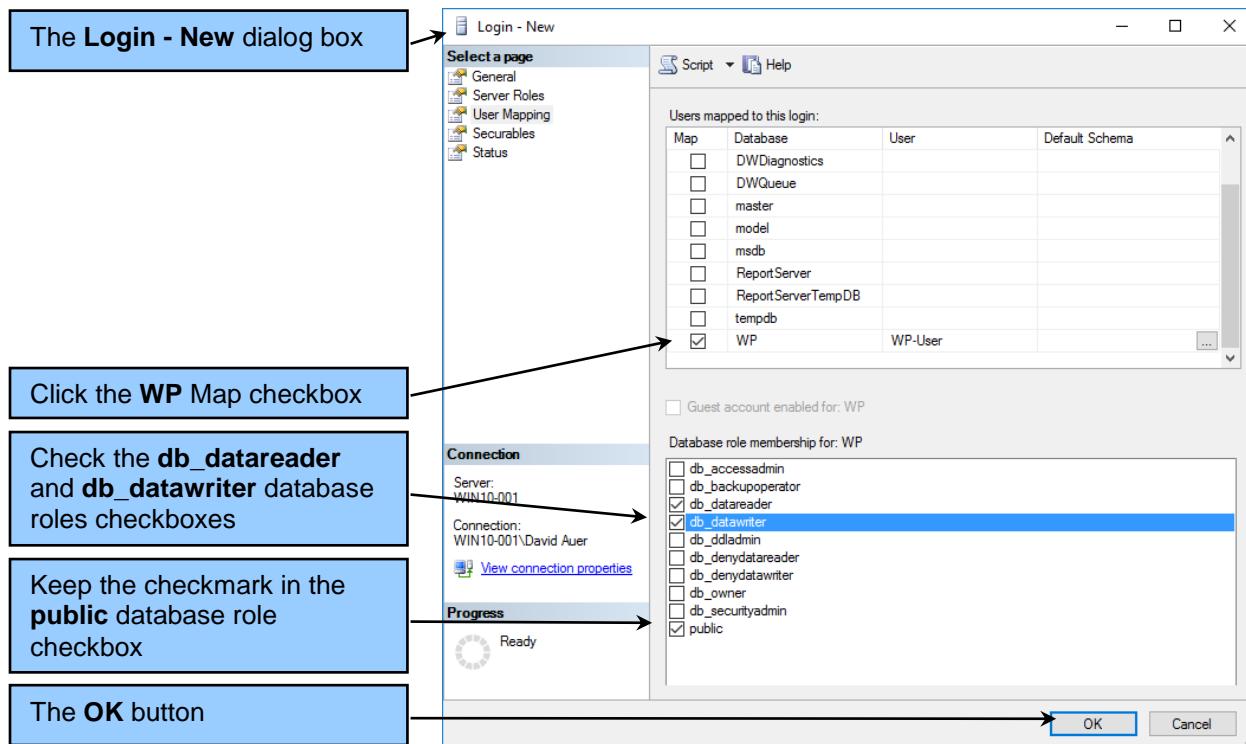


Figure A-45 — The Login – New Dialog Box User Mapping Page

13. Double check your settings, and then click the OK button shown in Figure A-45.
14. The new user login WP-User is created, and assigned the specified set of permissions to the WP database.

How Do I Create an ODBC Connection from Microsoft Access 2016 to an SQL Server 2016 Database?

While SQL Server 2016 is an excellent enterprise-class DBMS, it does not provide any application development tools. Microsoft Access 2016 does provide a set of application development tools such as forms, reports, stored queries, and menu systems (see Appendix H, “The Access Workbench—Section H—Microsoft Access 2016 Switchboards”). Thus it would be useful to have a way to use Microsoft Access 2016 as the application development frontend for an SQL Server 2016 database.

This is actually very easy to do using an **Open Database Connectivity (ODBC)** link. For a full discussion of ODBC and how to use it in Web-based database applications, see Chapter 7. Here, we will simply walk through the steps necessary to build and use the connection.

We will continue to use the WP database that we have been using, and we will create a Microsoft Access 2016 database to act as the application development environment for the WP database. We will name our Microsoft Access 2016 database as WPIS.accdb (for Wedgewood Pacific Information System.)

Creating the WPIS.accdb Database:

1. Open **Microsoft Access 2016**.
2. Click the **Blank desktop database** icon to open the Blank desktop database dialog box.
3. In the **Blank desktop database** dialog box, type in the file name **WPIS.accdb**, and then click the **Create** button.
4. The new WPIS.accdb database is displayed.
5. Close the open **Table1** tabbed window.

Because we will be connecting to a DBMS that uses **ANSI standard SQL (ANSI 92)** instead of **Microsoft ANSI-89 SQL** (this topic is discussed in depth at the beginning of Chapter 3), we need to set the Microsoft Access 2016 options for WPIS.accdb to working with ANSI standard SQL.

Setting the Microsoft Access 2016 SQL Setting:

1. Click the **File** command tab, and then click the **Options** button. The Access Options dialog box is displayed.
2. In the **Access Options** dialog box, click the **Object Designers** button to display the Object Designers page.
3. In the **Object Designers** page, find the **SQL Server Compatible Syntax (ANSI 92)** section of the Query Design settings. Click the **This database** checkbox as shown in Figure A-46.
4. Click the **OK** button to save the settings and close the dialog box.

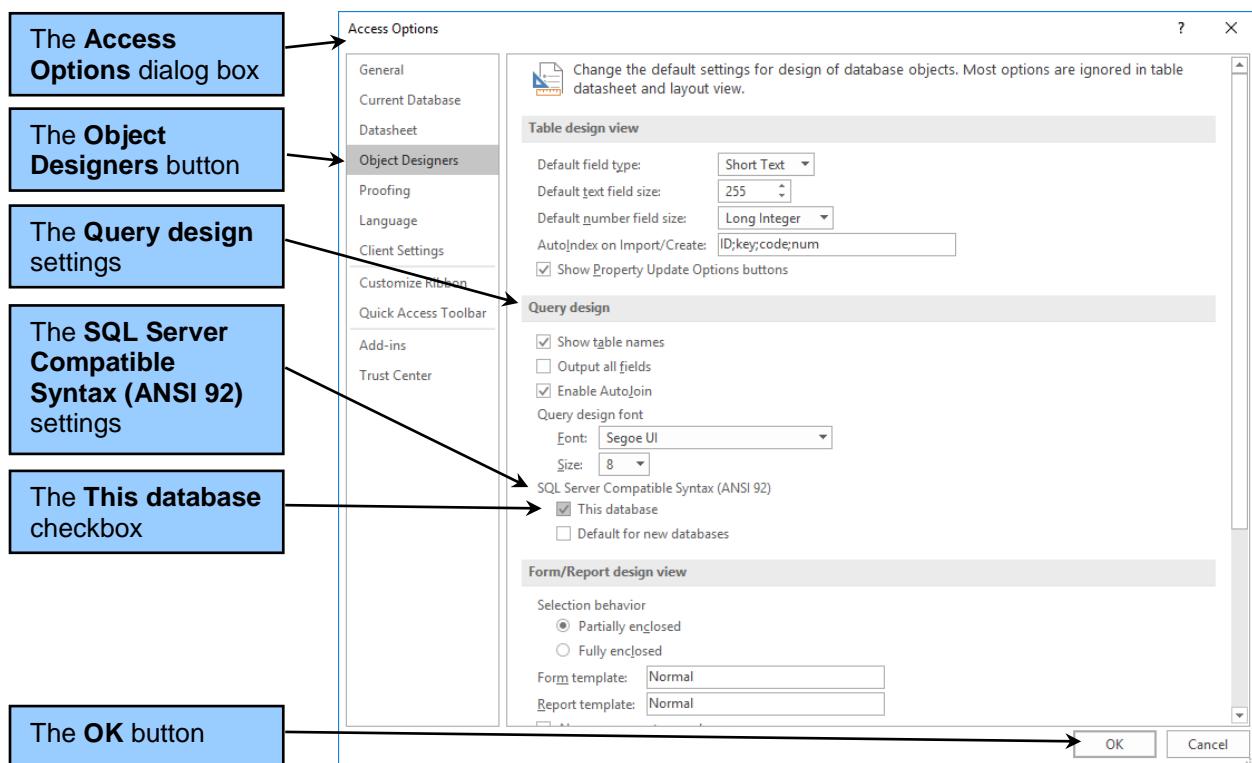


Figure A-46 — The SQL Server Compatible Syntax (ANSI 92) Checkbox

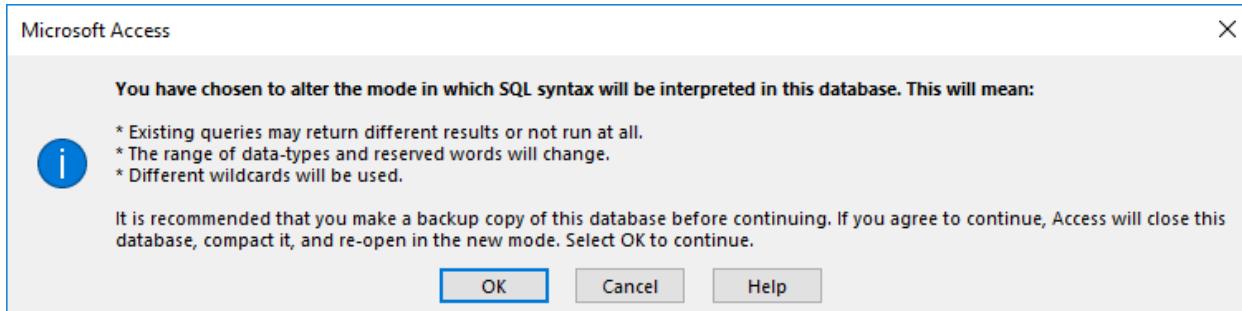


Figure A-47 — The Microsoft Access Information Dialog Box

5. A Microsoft Access Information dialog box is displayed, as shown in Figure A-47.
6. Read the information in the dialog box, and then click the **OK** button
7. Microsoft Access take the actions discussed in the information dialog box, and then reopens the WPIS.accdb database with the **Security Warning message bar** displayed.
8. In the Security Warning message bar, click the **Enable Content** button.

The WPIS database is now ready to use. Our next step is to connect to the WP database in SQL Server 2016. To do this we create a link to data external to Microsoft Access by using a ODBC data source. For a full discussion of ODBC data sources (each of which is called a **DSN**), see Chapter 7. And in creating the ODBC DSN, we will make use of the WP-User login we previously created.

Linking the Microsoft Access 2016 Database to an External Data Source via ODBC:

1. In the Microsoft Access 2016 WPIS.accdb database, click the **External Data** command tab, and then click the **ODBC Database** button in the Import & Link commands section.
2. The **Get External Data - ODBC Database Wizard** dialog box is displayed, as shown in Figure A-48.
3. In the Get External Data – ODBC Database Wizard dialog box, the *Select the source and destination of the data* page is displayed. Click the **Link to the data source by creating a linked table** radio button as shown in Figure A-48, and then click the **OK** button.
4. The **Select Data Source** dialog box is displayed, as shown in Figure A-49. This is the dialog box that we will use to create the needed ODBC DSN.
5. In the Select Data Source dialog box, make sure the File Data Source tab is selected, and then click the **New** button to display the **Create New Data Source** dialog box as shown in Figure A-50.
6. In the Create New Data Source dialog box, scroll down through the list of drivers until you can see the driver named **ODBC Driver 13 for SQL Server**. Click this driver name to select it, and then click the **Next** button.

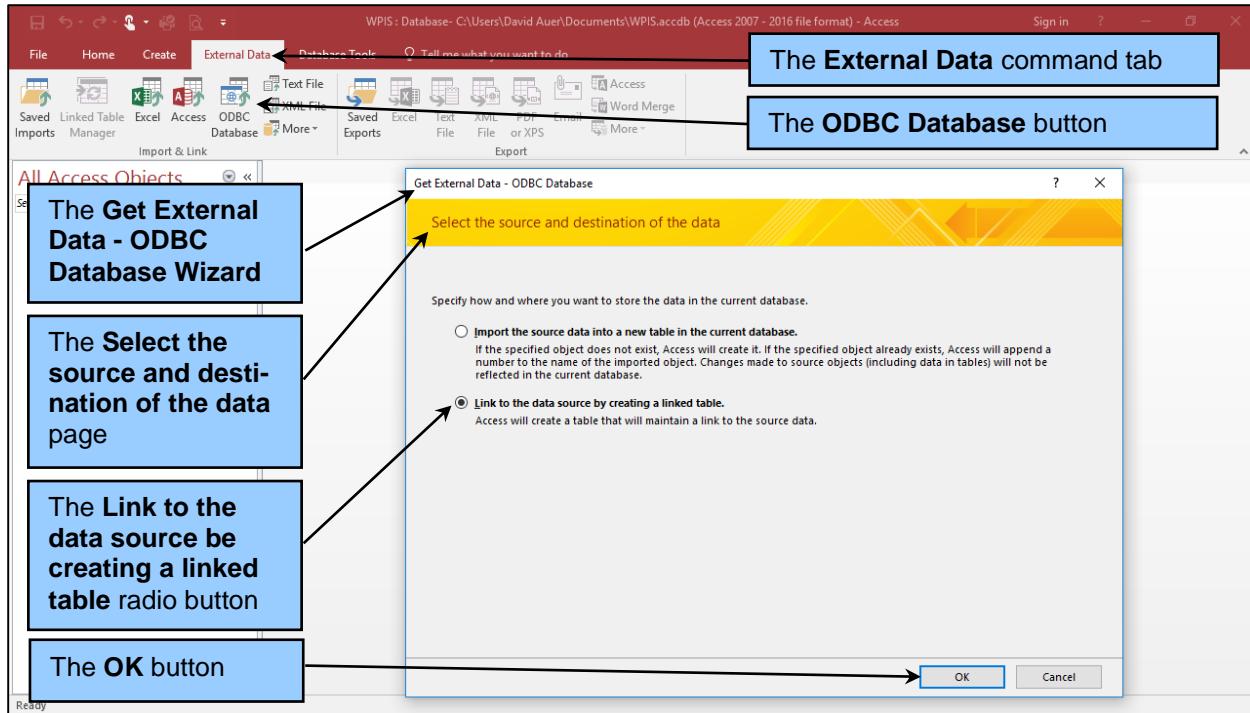


Figure A-48 — The Get External Data - ODBC Database Wizard Dialog Box

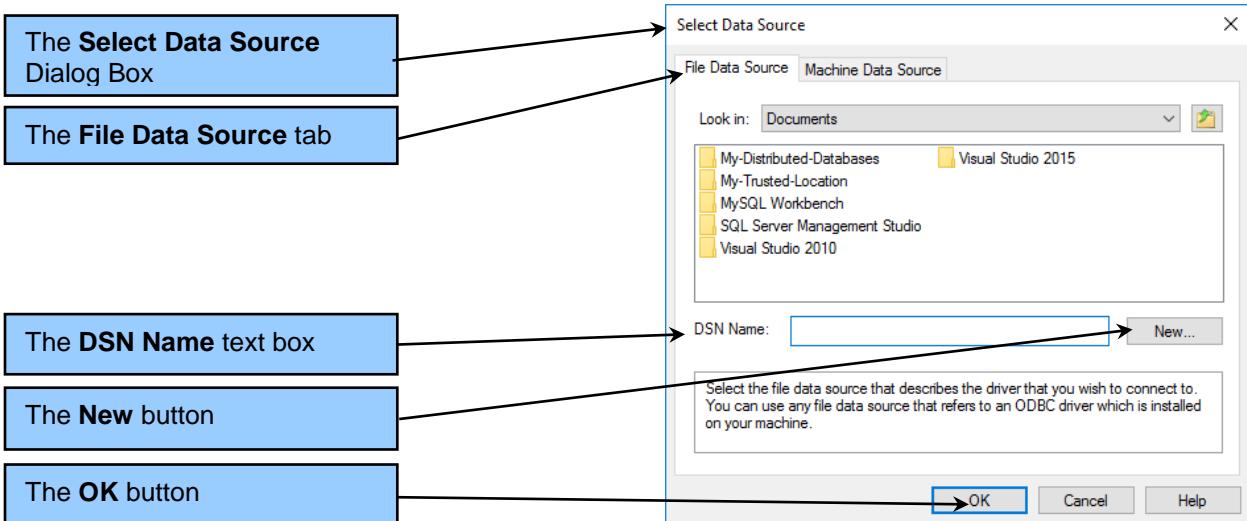


Figure A-49 — The Select Data Source Dialog Box

7. As shown in Figure A-51, the next page of the Create New Data Source dialog box provides a text box for naming the new DSN. Type in **WP**, and then click the **Next** button.
8. As shown in Figure A-52, the next page of the Create New Data Source dialog box provides a summary of the settings that will be used to create the new DSN. Click the **Finish** button.

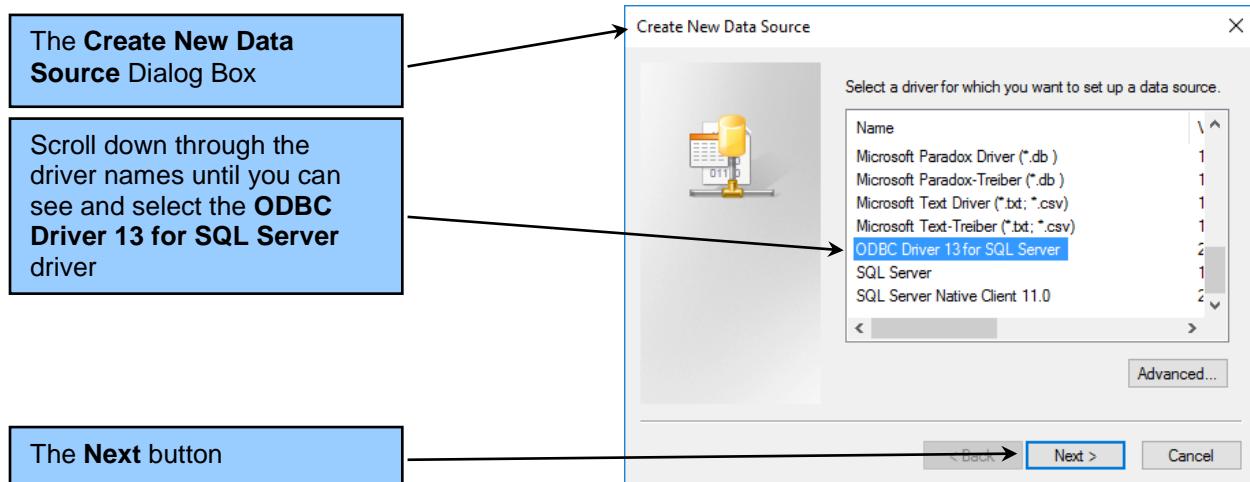


Figure A-50 — The Create New Data Source Dialog Box

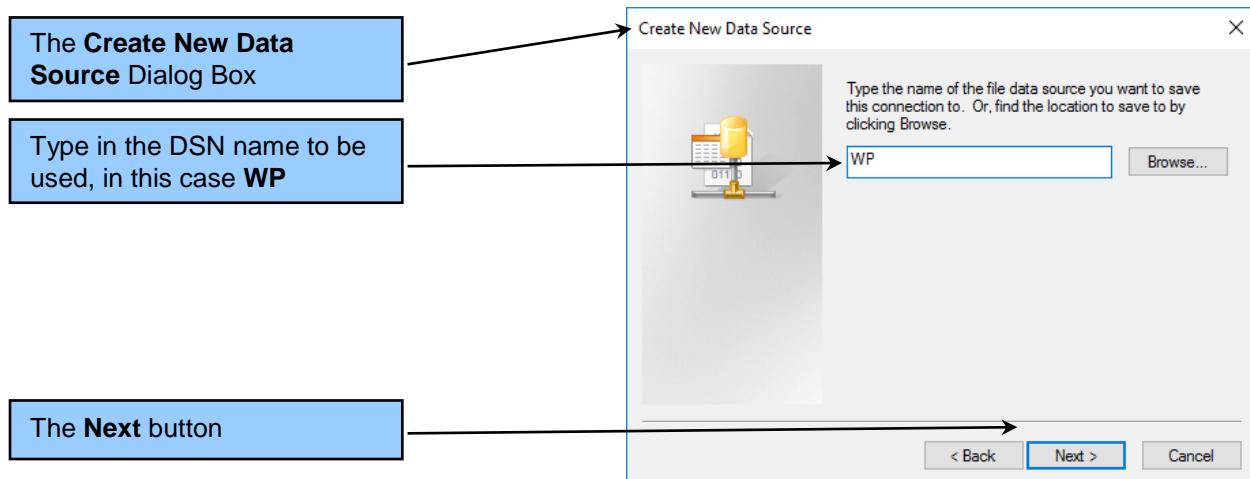


Figure A-51 — Naming the DSN

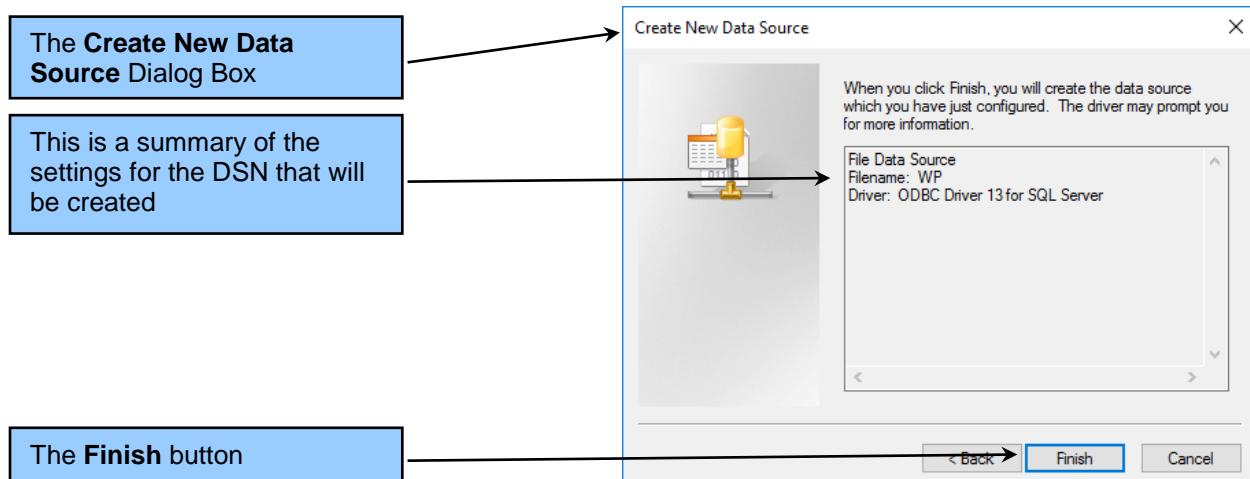


Figure A-52 — The Create New Data Source Dialog Box

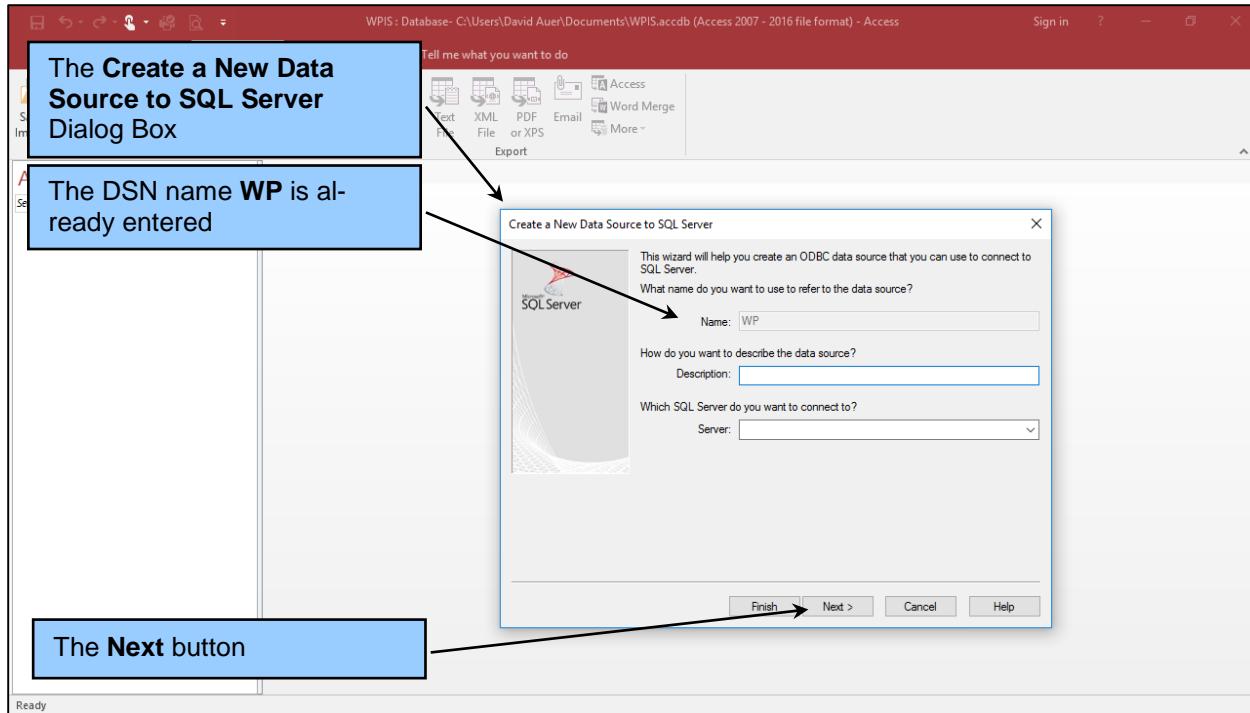
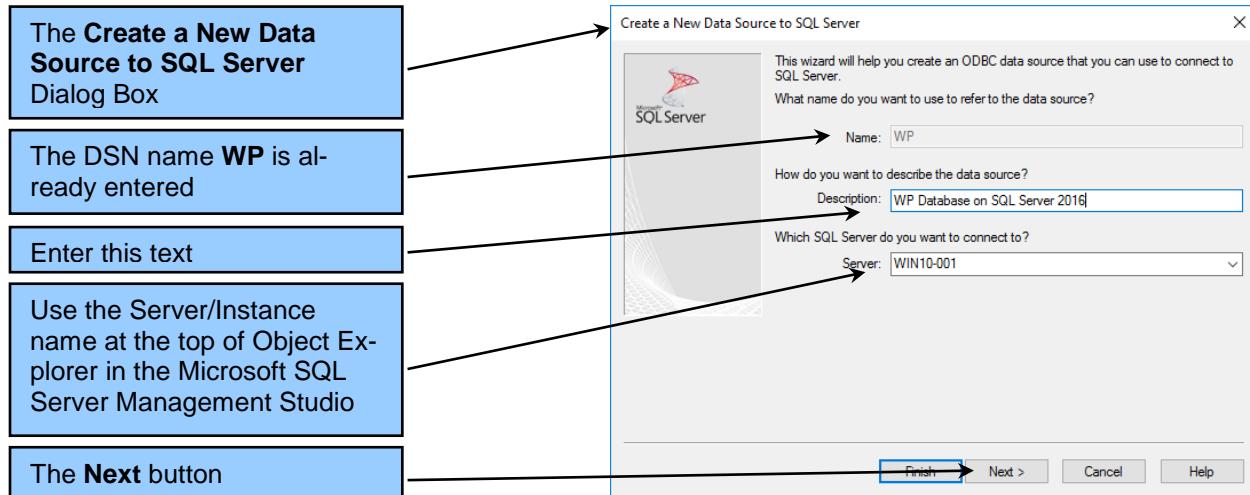
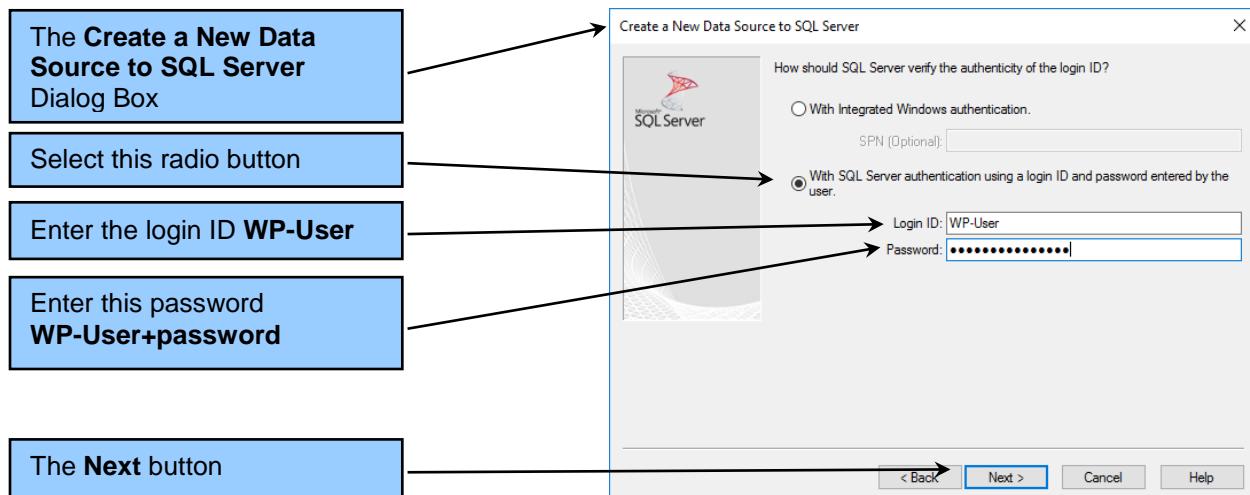
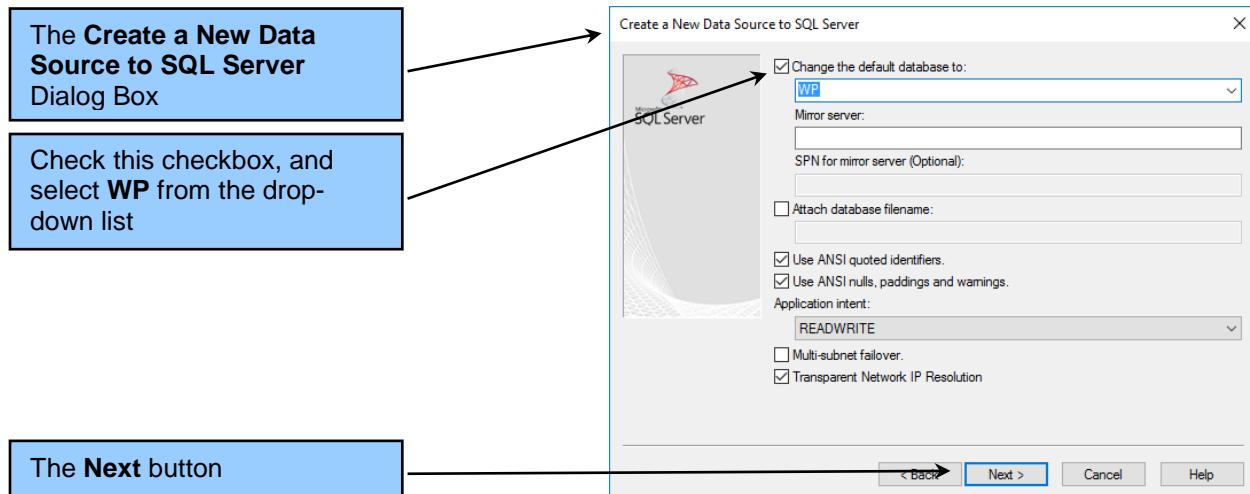


Figure A-53 — The Create a New Data Source to SQL Server Wizard Dialog Box

9. The **Create a New Data Source to SQL Server** Wizard dialog box is displayed, as shown in Figure A-53. The DSN name of WP is already assigned.
10. In the **Description** text box enter the text **WP Database on SQL Server 2016**.
11. In the **Server** text box, type in the **SQL Server name** exactly as it appears at the top of the Object Explorer window in the SQL Server Management Studio. In Figure A-42, you can see that our server name is *WIN10-001*, so this is what we will enter. This name may consist of two parts—a *computer name* (in our case WIN10-001) and an SQL Server 2016 *instance name*. If you have only one instance of SQL Server 2016 on your computer, and it was installed with the default instance name of MSSQLSERVER, then *no* instance name appears in Object Explorer, and *no* instance name is needed in the Server text box. This is what has happened in our case.
12. The dialog box now appears as shown in Figure A-54.
13. Click the **Next** button. The Login settings page is displayed. Click the **With SQL Server authentication using a login ID and password entered by the user** radio button, and then enter the login ID **WP-User** and the password **WP-User+password**.
14. The dialog box now appears as shown in Figure A-55.
15. Click the **Next** button. A database settings page is displayed. Set the default database to **WP**, but leave all the other setting as they are.
16. The dialog box now appears as shown in Figure A-56.

**Figure A-54 — The Completed Server Settings****Figure A-55 — The Completed Login Settings****Figure A-56 — The Completed Database Settings**

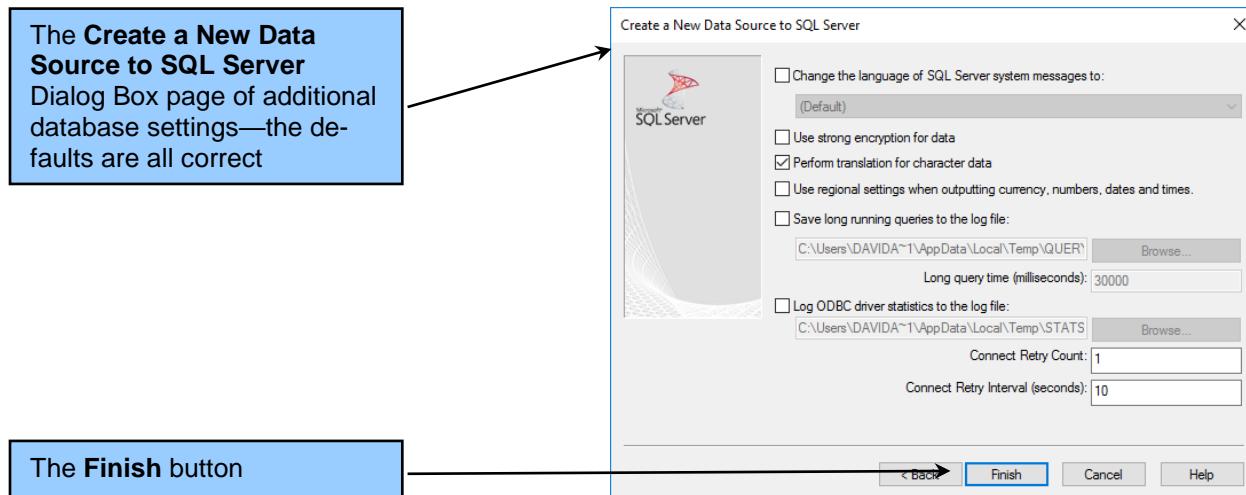


Figure A-57 — The Additional Database Settings Page

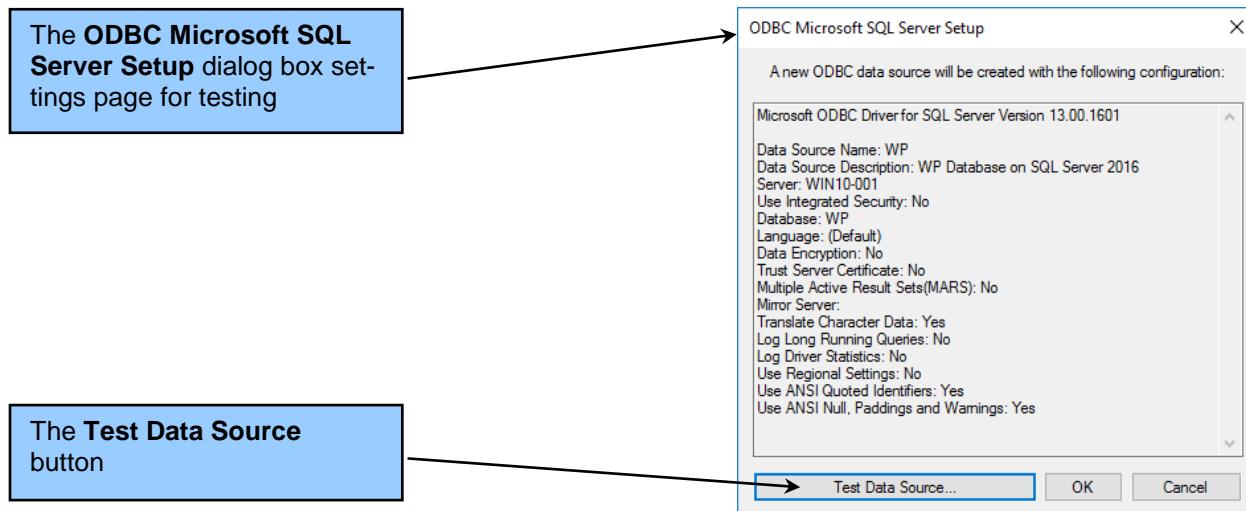


Figure A-58 — The ODBC Microsoft SQL Server Setup Page

17. Click the **Next** button. An additional set of database settings page is displayed, as shown in Figure A-57. The defaults shown here are correct. Click the **Finish** button.
18. The **ODBC Microsoft SQL Server Setup** dialog box is displayed, showing the settings that will be used to create the DSN. Additionally, the dialog box has a **Test Data Source** button that can be used to test the DSN before it is created.
19. Click the **Test Data Source** button. If all the settings are correct, the **SQL Server ODBC Data Source Test** dialog box is displayed with the message "TESTS COMPLETED SUCCESSFULLY" as shown in Figure A-59.
20. In the **SQL Server ODBC Data Source Test** dialog box, click the **OK** button.

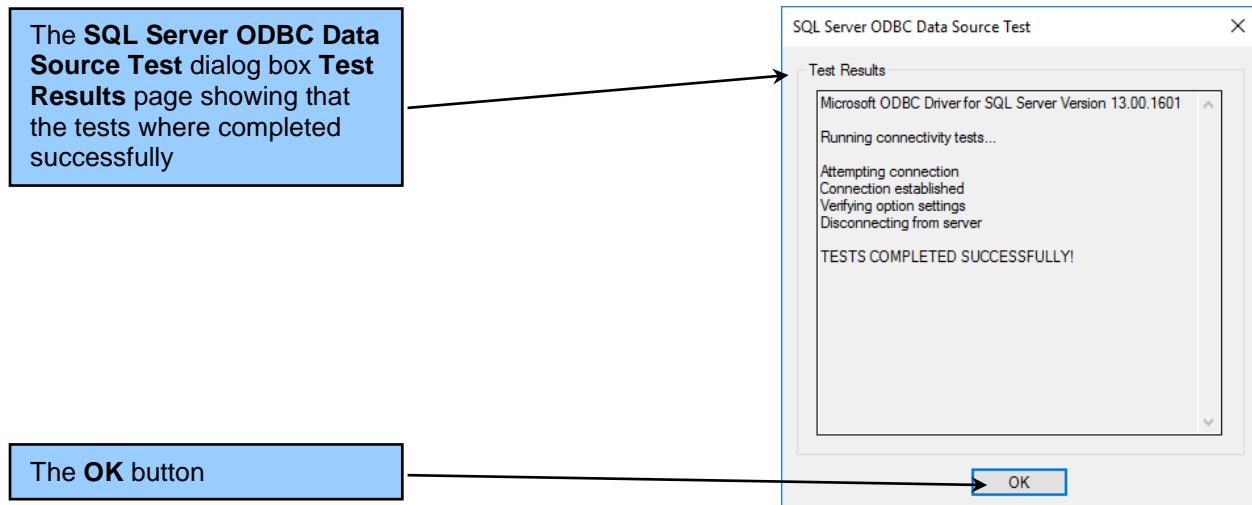


Figure A-59 — The SQL Server ODBC Data Source Test Dialog Box

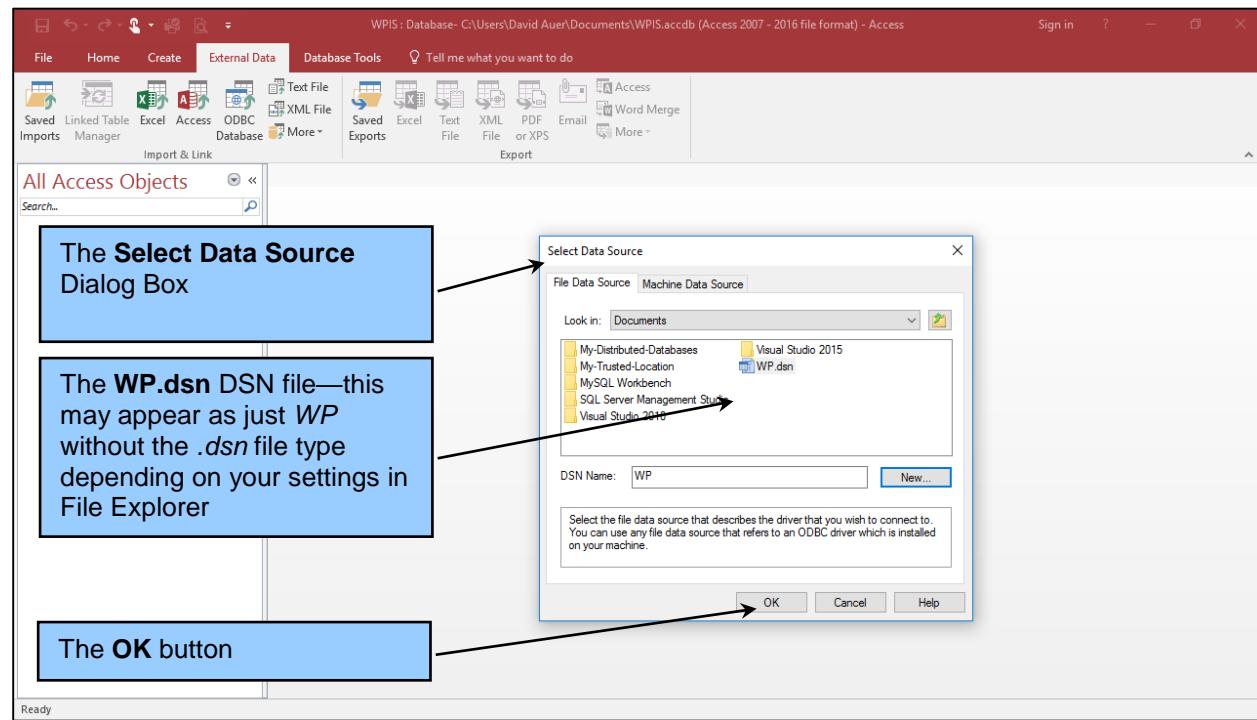


Figure A-60 — The Completed WP DSN

21. The **ODBC Microsoft SQL Server Setup** dialog box is displayed. Click the **OK** button.
22. The WP DSN file data source is created and displayed in the Select Data Source dialog box as shown in Figure A-60. Now that the DSN is completed, we can finish linking the Microsoft Access 2016 database to the SQL Server database.
23. In the **Select Data Source** dialog box, click the **OK** button.

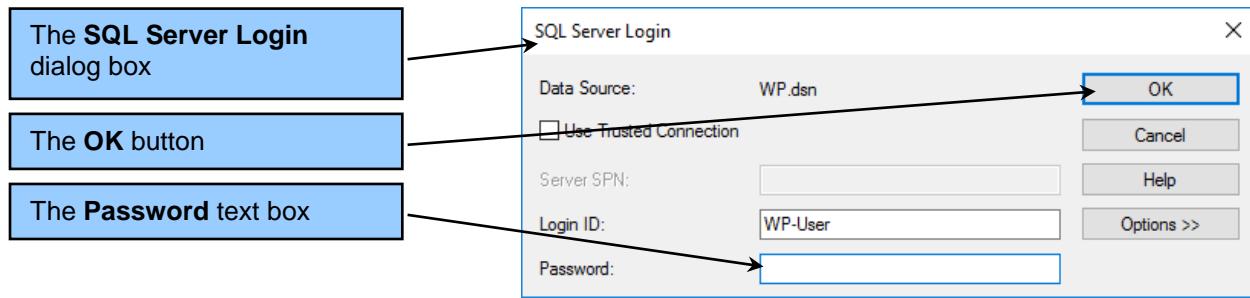


Figure A-61 — The SQL Server Login Dialog Box

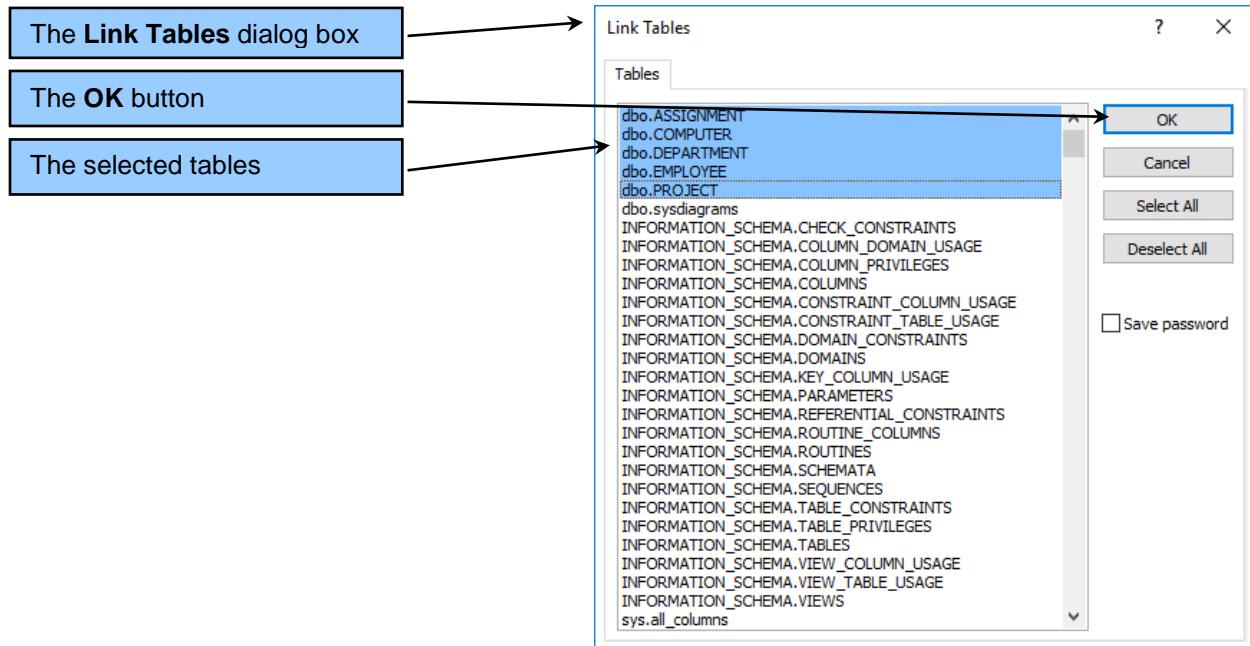


Figure A-62 — The Link Tables Box

24. The **SQL Server Login** dialog box is displayed, as shown in Figure A-61. Enter the password **WP-User+password** in the **Password** text box, and then click the **OK** button.
25. The **Link Tables** dialog box is displayed, as shown in Figure A-62. Note that five tables that have been selected. To select the **dbo.ASSIGNMENT** table, **click on it**. To add each additional table to the selection, **press the Ctrl key and then click on the table name**. *dbo* stands for *database owner* and is commonly shown in SQL Server object names (technically, it is a *schema* name).
26. Click the **OK** button. The ODBC links between the WP.accdb Microsoft Access 2016 and the WP database in SQL Server 2016 are completed, as shown in Figure A-63.
27. We can now use the WP.accdb database to create application tools such as forms and reports. For example, a form to display, edit, insert, and delete WP employee data is shown in Figure A-64. The data that is shown in this form is actually in the SQL Server 2016 WP database.
28. Close the WP.accdb Microsoft Access database.

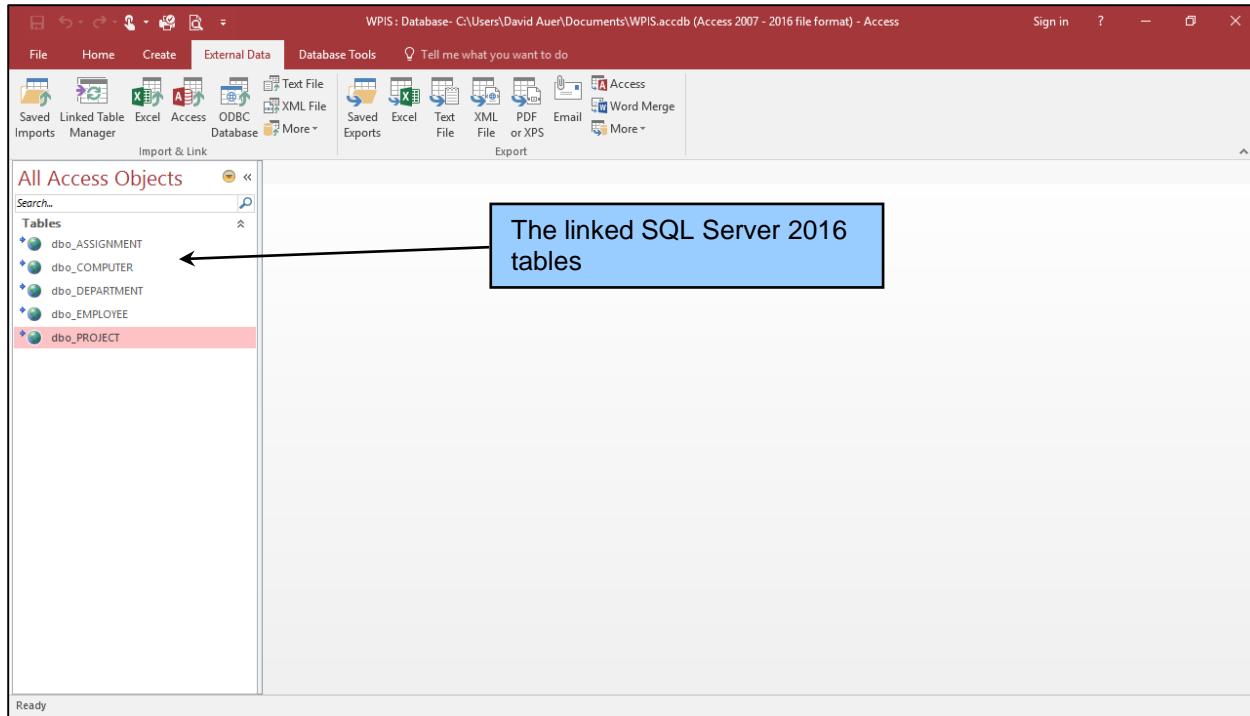


Figure A-63 — The Linked SQL Server 2016 Tables in the Microsoft Access Database

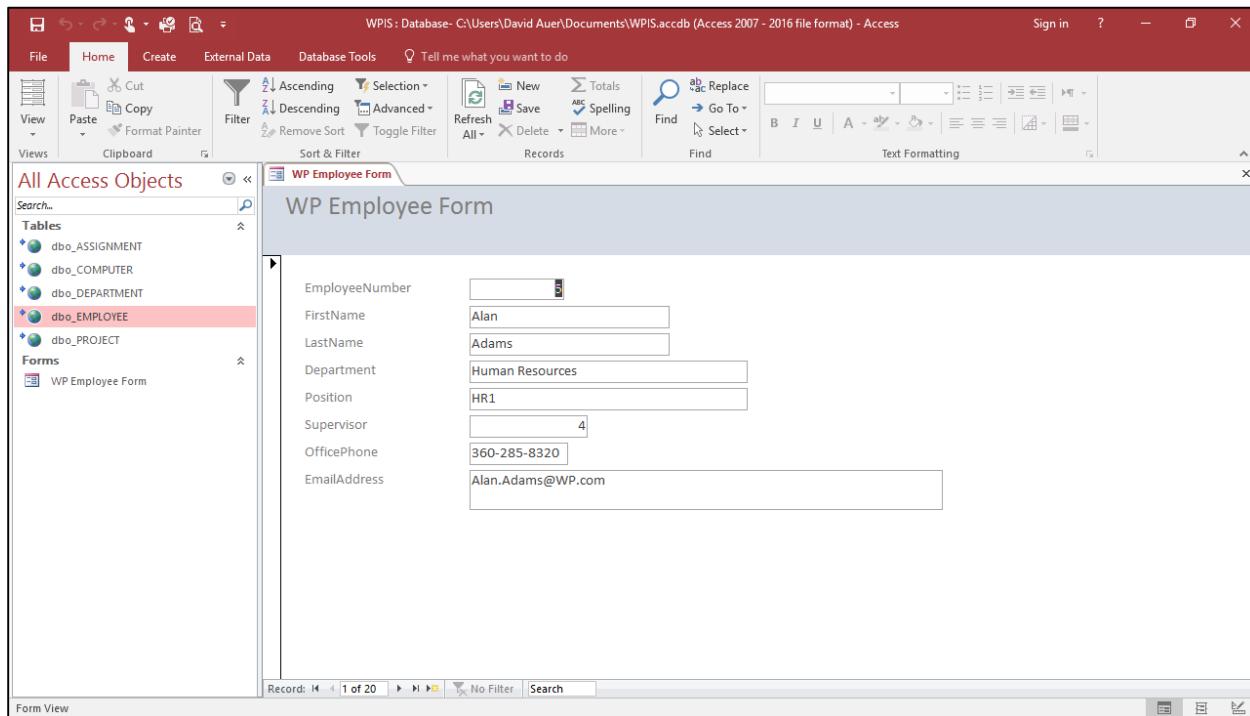


Figure A-64 — The WP Employee Form

Additional Documentation for SQL Server Express Edition

To get access to SQL Server documentation, use **SQL Server 2016 Books Online** at
<http://msdn.microsoft.com/en-us/library/ms130214.aspx>.

KEY TERMS

*.sql	Active database
ANSI standard SQL (ANSI 92)	Available Databases drop-down list
bulk INSERT statement	Business Intelligence Edition
Close button	Connect button
Connect to Server dialog box	Critical update for SQL Server 2016 MSVCRT prerequisites
Databases (folder)	Developer Edition
DSN	Enterprise Edition
Execute button	Express Edition
Express Advanced	extract, transform and load (ETL)
IntelliSense Enabled button	Messages window
Microsoft Access Database Engine 2010 Redistributable	Microsoft ANSI-89 SQL
Microsoft SQL Server	Microsoft SQL Server 2016 Management Studio
Microsoft .NET Framework 3.5 Service Pack 1	Microsoft Windows Installer 4.6.1
Mixed (authentication method)	multivalue, multicolumn problem
multivalued dependency	New Database dialog box
New Query button	Object Explorer (window)
Oracle Java Runtime Environment (JRE)	Open Database Connectivity (ODBC)
Parse button	PC Documents (folder)
Projects (folder)	Results window

ReportServer	ReportServerTempDB
SQL Editor toolbar	SQL Server 2005 Express Edition
SQL Server 2008	SQL Server 2008 R2
SQL Server 2012	SQL Server 2014
SQL Server 2016	SQL Server 2016 Express
SQL Server 2016 Express Advanced	SQL Server 2016 Report Services
SQL Server 2016 Management Studio	SQL Server 2016 Books Online
SQL Server Import and Export Wizard	SQL script
spreadsheets	Standard Edition
Standard toolbar	Update for Visual C++ 2013, Visual C++ Redistributable Package, and KB3164398
Windows Authentication (authentication method)	worksheets
X32	X64

REVIEW QUESTIONS

- A.1 What is SQL Server 2016 Developer edition? What is SQL Server 2016 Express edition? What are the differences between these two editions of SQL Server 2016?
- A.2 What is the primary advantage of using SQL Server 2016 instead of Microsoft Access?
- A.3 What set of programs is recommended as a necessary set of prerequisites to the actual installation of SQL Server 2016? In what order should you install these products?
- A.4 How do you install Microsoft SQL Server 2016 Developer edition?
- A.5 What is the purpose of the Microsoft SQL Server Management Studio?
- A.6 How do you install the Microsoft SQL Server Management Studio?
- A.7 If you experience a problem starting the Microsoft SQL Server Management Studio, what steps should you take to resolve this problem?
- A.8 How do you create a new database in SQL Server 2016?
- A.9 How do you specify the active database in SQL Server 2016?

- A.10 What is an SQL script? What types of SQL statements and commands can you run more efficiently as scripts?
- A.11 What tool(s) can be used to create a script?
- A.12 What file extension should you use for SQL scripts?
- A.13 How do you open and run a script in SQL Server?
- A.14 How do you create database tables in SQL Server?
- A.15 How do you populate database tables in SQL Server?
- A.16 How do you create and run an SQL query in SQL Server?
- A.17 How do you install the Microsoft SQL Server 2016 ODBC client? Which ODBC software is installed in the process?
- A.18 How do you import Microsoft Excel data into an SQL Server table?
- A.19 Why should imported data be initially stored in a temporary table and then moved into a different SQL Server table? How do you do this?
- A.20 How do you create user accounts in SQL Server 2016? How do you give them appropriate permissions to a specific database?
- A.21 Why would you want to create an ODBC connection to link a Microsoft Access 2016 to an SQL Server Database?
- A.22 What is an ODBC DSN? Why is one needed?
- A.23 How do you create an ODBC connection to link a Microsoft Access 2016 database to an SQL Server Database?



EXERCISES

- A.24 If you haven't already done so, download and install SQL Server 2016 Developer edition as described in the text. Use the default settings for the installation. Be sure that the Microsoft SQL Server Management Studio is correctly installed.
 - A.25 If you haven't already done so, work through the steps described in this appendix to create and populate the WP database.
-

- A.26 Using SQL Server 2016 and the Microsoft SQL Server Management Studio, run the following SQL queries in Chapter 3:

SQL-Query-CH03-01 through SQL-Query-CH03-32

Skip SQL-Query-CH03-33 and SQL-Query-CH03-34 as they result in errors!

SQL-Query-CH03-35 through SQL-Query-CH03-51

Save each query as follows:

- Create and run each query in SQL Server Management Studio.
- After you have run each query, use the **File | Save SQLQuery#.sql As** command to save the query (The # sign in the name changes as you create different queries). By default, SQL Server saves each file as an SQL file with the file extension *.sql. Use this default setting unless your instructor tells you to use a different extension. Name your queries in *numerical sequence*, starting with the file name MSSQL-SQL-Query-CH03-01.sql.

- A.27 Use Microsoft SQL Server Management Studio to run one or more of the saved SQL queries you created in question A.26:

- Open a query using the **Open File** button (or by selecting the **File | Open File** menu command). Note that the query is opened in a tabbed query window. Run the query.
- Use the **Open File** button (or the **File | Open File** menu command) to open and run another query in another tabbed window.
- Experiment with opening and closing windows and running various queries in these windows.

- A.28 If needed, complete exercise 3.58. Complete exercise 3.59 using SQL Server 2016 and the Microsoft SQL Server Management Studio. Start each saved query name with MSSQL- and use the default .sql file extension. The first saved query name should be MSSQL-SQL-Query-AWE-3-1-A.sql.

- A.29 If needed, complete exercise 3.58. Complete exercise 3.60 using SQL Server 2016 and the Microsoft SQL Server Management Studio. Start the saved query name with MSSQL- and use the default .sql file extension. The first saved query name will be MSSQL-SQL-Query-AWE-3-3-A.sql.

- A.30 If you have not already done so, import the **COMPUTER** table from Microsoft Excel into the SQL Server 2016 WP database as explained in the text.

- A.31 Import the **COMPUTER_ASSIGNMENT** table from Microsoft Excel into the SQL Server 2016 WP database. Column characteristics are shown in Figure A-65 (this table shows the column characteristics with Microsoft Access 2016 data types). How should this table be linked to the **EMPLOYEE** table and the **COMPUTER** table by foreign keys? Be sure to include these foreign keys in your final **COMPUTER_ASSIGNMENT** table structure when you create it in SQL Server 2016.
-

Database Column Characteristics for the WP COMPUTER_ASSIGNMENT Table

Column Name	Type	Key	Required	Remarks
SerialNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
EmployeeNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
DateAssigned	Date/Time	Primary Key	Yes	Medium Date

Figure A-65 — Database Column Characteristics for the WP COMPUTER_ASSIGNMENT Table

- A.32 If you have not already done so, create the **WP-User** user account and associated permissions in the SQL Server 2016 WP database as explained in the text.
- A.33 Create a user account in the SQL Server 2016 WP database named **WP-Reader**. Give this user **SQL Server authentication** with the password of **WP-Reader+password** and with other password settings to match those shown in Figure A-44. Give WP-Reader a user mapping to the WP database with **public** and **db_datareader** permissions only.
- A.34 Create a Microsoft Access 2016 database named **WPIS_RO.accdb** where RO stands for “read-only.” This database will be a *read-only* application for the SQL Server 2016 WP database, which will allow users to read and query the data in the WP database but not to make any updates to the data or to insert new data. Then:
1. Set the WPIS_RO.accdb database to use **SQL Server Compatible Syntax (ANSI 92)**.
 2. Link the WPIS_RO.accdb database to the SQL Server 2016 WP database. When you create your File Data Source DSN, name the DSN **WP_RO**, and use the WP-Reader user account (as detailed in Exercise A.33) for SQL Server authentication.
 3. Import all existing tables (including the COMPUTER and COMPUTER_ASSIGNMENT tables if they have been imported as detailed in Exercises A.30 and A.31).
 4. Create a form to show all the data in PROJECT table named **WP Projects Form**.
 5. Create a report to show all the data in the PROJECT table named **WP Projects Report**.

