**Database Concepts**

8th Edition

### David M. Kroenke • David J. Auer • Scott L. Vandenberg • Robert C. Yoder

**Instructor’s Manual**

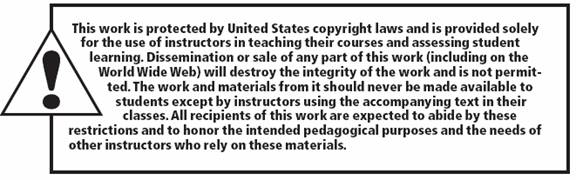
### Prepared by David J. Auer

**Appendix A**

**Getting Started with Microsoft SQL Server 2016**



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



Instructor’s Manual to accompany:

***Database Concepts (8th Edition)***

### David M. Kroenke • David J. Auer • Scott L. Vandenberg • Robert C. Yoder

© 2017, 2015, 2013, 2011, 2010, 2008 Pearson Education, Inc. Publishing as Prentice Hall

CHAPTER 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

CHAPTER ERRATA

These are no known errors at this time. Any errors that are discovered in the future will be reported and corrected in the online DBC e08 Errata document, which will be available at [http://www.pearsonhighered.com/kroenke](http://www.pearsonhighered.com/kroenke/).

THE ACCESS WORKBENCH

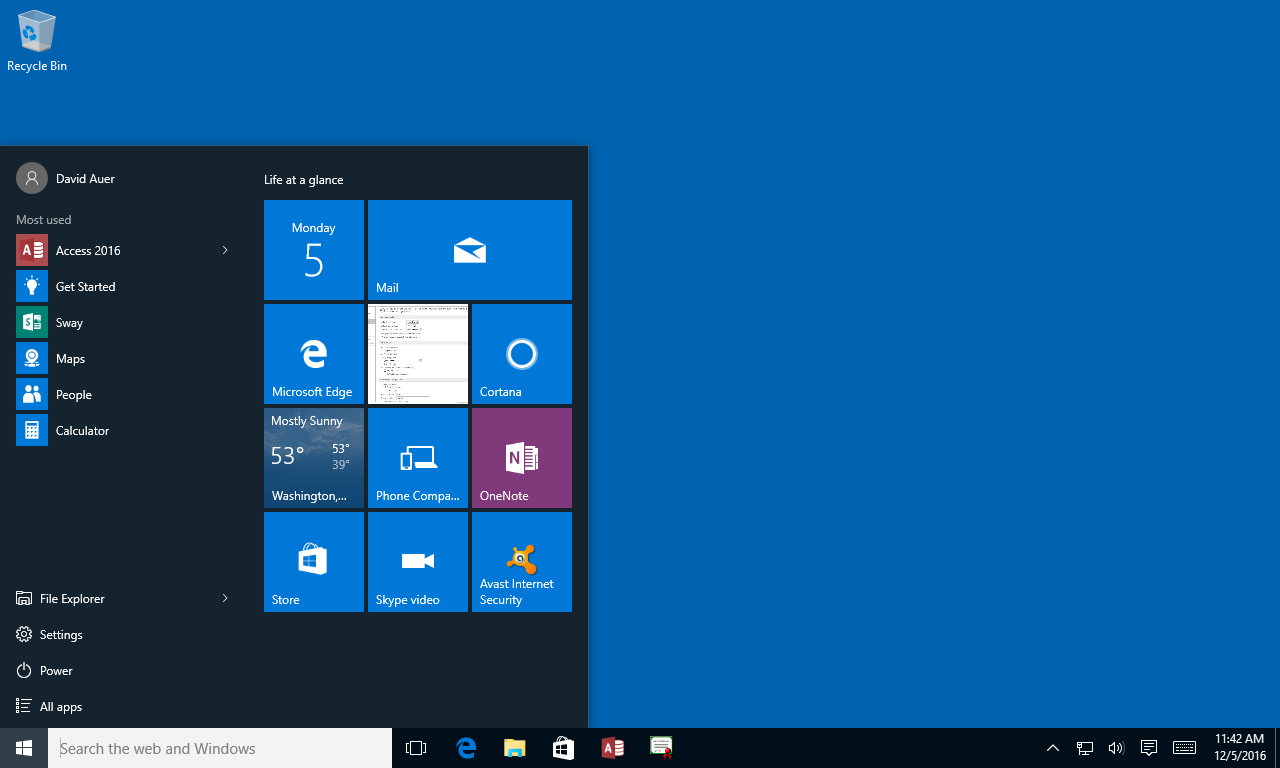
Solutions to the *Access Workbench* exercises may be found in *Solutions to all Sections:* *The Access Workbench*, which is a separate document within the Instructor’s Manual.

There is no section of *The Access Workbench* associated with this appendix.

NOTES ON MICROSOFT WINDOWS 10

This book uses the Microsoft Windows 10 operating system as the basis for screenshots and step-by-step instructions. However, with Windows 10, Microsoft has introduced a continuous update system that has already resulted in some fundamental differences in how different versions of Windows 10 look and operate.

For example, in the original version of Microsoft Windows 10, clicking the **Windows Start** button (or pressing the Windows key on the keyboard) displayed the menu shown in Figure 1. In this menu, we need to click the **All apps button** in order to see the **All apps menu** shown in Figure 2.

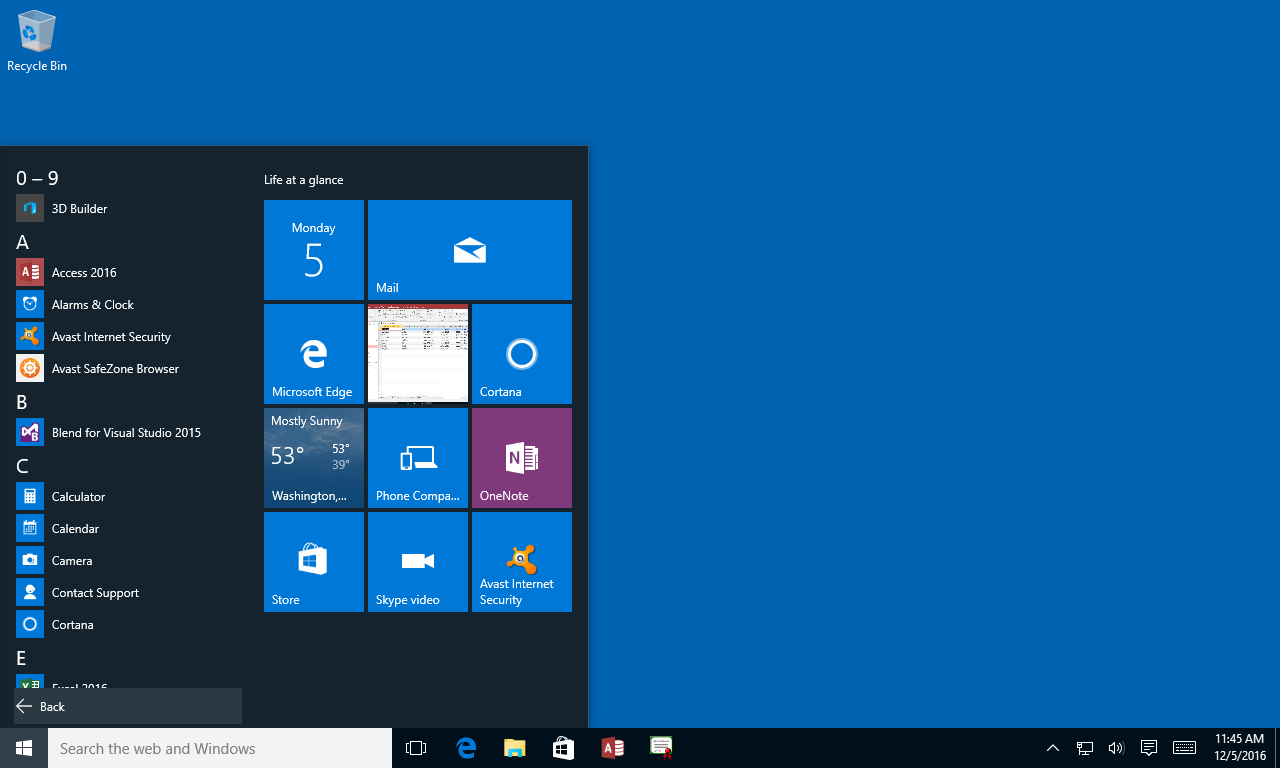


The **File Explorer** icon

The **All apps** button

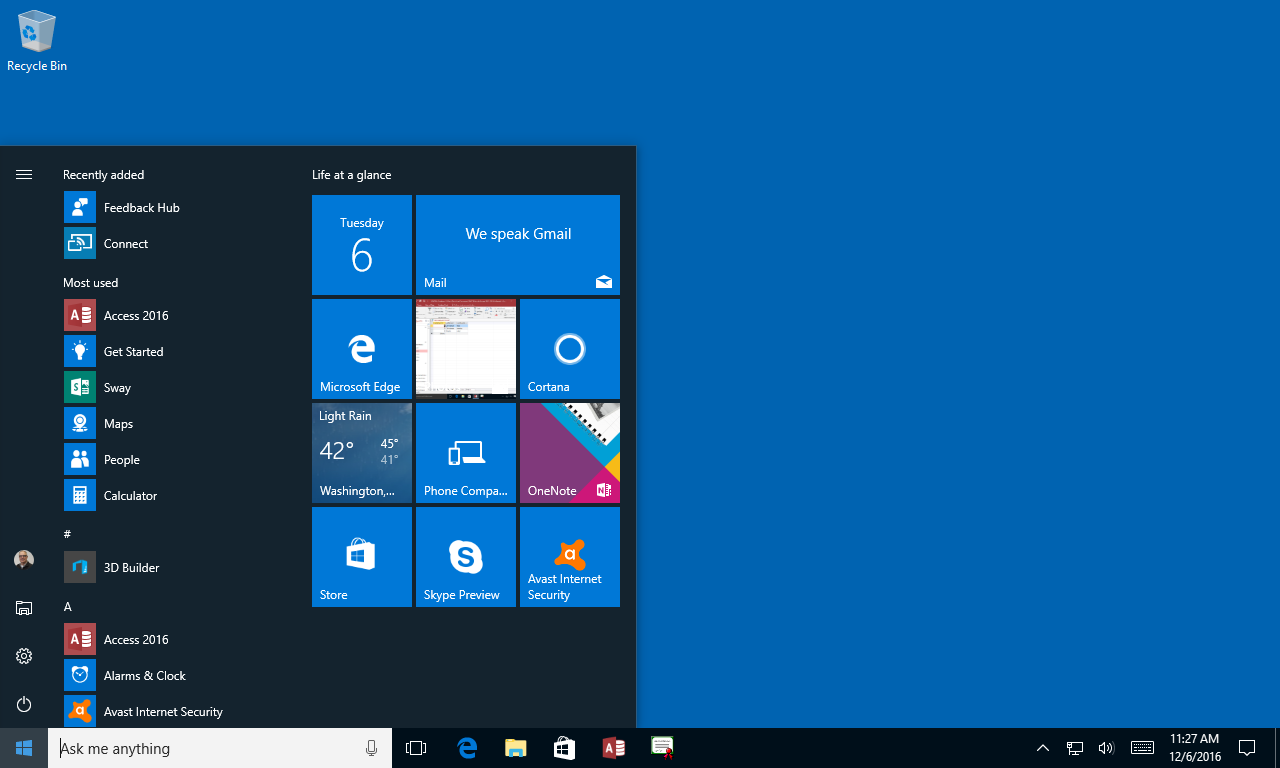
The **File Explorer** button

Figure 1 – Windows 10 Main Menu



The **All apps** menu

Figure 2 – Windows 10 All Apps Menu



The **File Explorer** icon

The **File Explorer** button

The **All apps** menu

Figure 3 – Windows 10 Anniversary Update Main Menu with All Apps Menu Included

Microsoft then released the Windows 10 Anniversary Update (Feature update to Windows 10, version 1607) (see the blog discussion at <https://blogs.windows.com/windowsexperience/2016/08/02/how-to-get-the-windows-10-anniversary-update/#K1CZuiw4auiuE9A5.97> ). One of the changes introduced in the Anniversary Update was a major change to the menu system. Now, as shown in   
Figure 3, the **All apps menu** is immediately available when the Start button is used (or when the keyboard Windows key is pressed).

Therefore, note that the step by step instructions in this book may need to be altered for your use depending upon which version of Microsoft Windows 10 you or your students are using!

We recommend that you update Windows 10 to the Windows 10 Anniversary Update (Feature update to Windows 10, version 1607), and make sure it is patched with all updates to that version (at a minimum patched to Windows 10 Version 1607 update for August 23, 2016 (KB3176936), and the Windows 10 Version 1607 cumulative update for September 29, 2016 (KB3194496). We also recommend using the 32-bit version of Microsoft Office. This insures that all the examples discussed in this book will function properly.

TEACHING SUGGESTIONS

* For individual student personal use on their own computer, we recommend (and use in this book) Microsoft SQL Server 2016 Developer edition. Microsoft SQL Server 2016 Express Advanced edition can be used as an alternative.
* When you are using Microsoft SQL Server 2016, the best text editor to use is the text editor built into the Microsoft SQL Server Management Studio. Take some time to show you students how to use it.
* Make sure that your students work through Appendix A in conjunction with other material presented in the text by working through Appendix A in the following sequence:
* Before starting Chapter 3 on SQL, install Microsoft SQL Server 2016 and Microsoft SQL Server Management Studio by working through this appendix up to and including “How Do I Install the Microsoft SQL Server Management Studio?”
* When studying 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.
* When studying 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.
* When studying the Appendix E material on SQL Views, you can run the SQL Statements shown in Appendix E yourself and see the results.
* When studying 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 t 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.

ANSWERS TO REVIEW QUESTIONS

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?

**Developer Edition** 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.

**Enterprise Edition** 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.

**Express Edition** is 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.

The differences are:

* Developer edition can be used by only one person in a non-production environment, while Express edition can be used in a multi-user, production environment.
* Developer edition supports more CPUs or core, larger computer memory and storage.
* Developer edition is full featured, while Express (and even Express Advanced) has feature limitations.

1. What is the primary advantage of using SQL Server 2016 instead of Microsoft Access?

SQL Server 2016 handles SQL much better than Microsoft Access.

1. What is the set of Microsoft programs that are recommended as a necessary set of prerequisites to the actual installation of SQL Server 2016? In what order should you install these products?
   1. Microsoft .NET Framework 3.5 SP 1
   2. Microsoft .NET Framework 4.6.1
   3. Oracle Java Runtime Environment (JRE) 7 Update 51 or higher
   4. Update for Visual C++ 2013, Visual C++ Redistributable Package, and KB3164398
   5. Microsoft SQL Server 2016 Developer Edition
   6. Microsoft SQL Server Management Studio
   7. Critical update for SQL Server 2016 MSVCRT prerequisites.

Install these in the order listed. As Microsoft incorporates updates into SQL Server 2016, it may become unnecessary to separately install the *Critical update for SQL Server 2016 MSVCRT prerequisites*.

1. How do you install Microsoft SQL Server 2016 Developer edition?

Install Microsoft SQL Server 2016 Developer as follows (this is an outline – we will not repeat the full set of instructions contained in the Appendix). Note that we are actually downloading SQL Server 2016 with Service Pack 1:

* 1. Download **SQLServer2016-SSEI-Dev.exe**, which is a stub that starts the download of the actual installation file.
  2. Download **SQLServer2016SP1-FullSlipStream-x64-ENU-DEV.iso**, which is the actual installation file in a format that is used to burn a DVD or be mounted as a DVD for installation.
  3. Mount **SQLServer2016SP1-FullSlipStream-x64-ENU-DEV.iso**.
  4. In the mounted file set, right-click the **setup.exe** file, and then click the **Run as administrator** command,
  5. In the **SQL Server Installation Center** dialog box, click the **Installation** button.
  6. In the Installation screen, click **New SQL Server standalone installation or add features to an existing installation**.
  7. When the **SQL Server 2016 Setup** dialog box is displayed, follow the steps detailed in the text and Figure A-8.

1. What is the purpose of Microsoft SQL Server Management Studio?

The Microsoft SQL Server Management Studio is the graphical management utility for SQL Server 2016. Using SQL Server Management Studio makes it much easier to work with SQL Server 2016. It is used to create databases and SQL scripts, and to run SQL scripts and commands.

1. How do you install the Microsoft SQL Server Management Studio?

Install the Microsoft SQL Server Management Studio as follows (this is an outline – we will not repeat the full set of instructions contained in the Appendix). Note that SQL Server Management Studio now requires a separate download:

* 1. Open the **SQL Server Installation Center** dialog box, click the **Installation** button.
  2. In the Installation screen, click **Install SQL Server Management Tools**.
  3. When the **Download SQL Server Management Studio (SSMS)** Web page is displayed, click the **Download SQL Server Management Studio {VersionNumber}** button and down load the file to the **This PC | Downloads** folder.
  4. Open File Explorer and browse to the **This PC | Downloads** folder. 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.
  5. The **Microsoft SQL Server Management Studio installation** dialog box is displayed. Click the **Install** button to begin the installation process.
  6. When the installation is complete, the Setup Completed page is displayed. Click the **Close** button to complete the installation process.
  7. Close the **SQL Server Installation Center** Dialog Box.

1. If you experience a problem starting the Microsoft SQL Server Management Studio, what steps should you take to resolve this problem?

Microsoft SQL Server 2016 may either take a long time to start the required services, or may fail to start the at all (we do *not* know why this occurs, and hope that Microsoft will fix this glitch soon!)

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

1. How do you create a new database in SQL Server 2016?

To create an SQL Server database, right-click the object in the Object Explorer to display the shortcut menu, and then click New Database.

1. How do you specify the active database in SQL Server 2016?

Specify the database in the SQL Query Toolbar by selecting the database name from the  
drop-down list.

1. What is a SQL script? What types of SQL statements and commands can you run more efficiently as scripts?

A SQL script is a related group of SQL statements intended to be run at the same time. Scripts are efficient for processing groups of SQL statements such as:

* 1. A set of CREATE TABLE commands to build a new database structure.
  2. A set of INSERT commands when data needs to be added to a table.

1. What tool(s) can be used to create a script?

Scripts can be created in Microsoft SQL Server Management Studio or in any other ASCII text editor, such as the Windows Notepad text editor. Microsoft SQL Server Management Studio is the recommended tool for script creation and editing.

1. What file extension should you use for SQL scripts?

The file extension *.sql* should be used so that such files are recognizable by the Microsoft SQL Server Management Studio and SQL Server 2016.

1. How do you open and run a script in SQL Server?

To open a script in SQL Server Management Studio Express, use the **File | Open | File . . .** menu command. In the **Open File** dialog box, select the **script file name** and then click the **Open** button. At this point, SQL Server *may* ask you to authenticate again. If so, a dialog box named **Connect to Database Engine** will appear, and you will have to click the **Connect** button. The script appears in a tabbed window labeled with the name of the script. To run the script, first specify the database to be used, and then click the **Execute** button in the **SQL Editor toolbar**.

After the script is executed, a Message window appears below the script window indicating success (or displaying appropriate error messages).

1. How do you create database tables in SQL Server?

To create database tables, create an SQL script containing the necessary SQL CREATE TABLE statements. Save and name the script. Be sure the correct database has been selected in the drop-down database list, and/or include the following code at the start of your script:

USE {DatabaseName}

GO

Run the script.

1. How do you populate database tables in SQL Server?

To create populate database tables, create an SQL script containing the necessary SQL INSERT statements. Save and name the script. Be sure the correct database has been selected in the drop-down database list, and/or include the following code at the start of your script:

USE {DatabaseName}

GO

Run the script.

1. How do you create and run an SQL query in SQL Server?

To run a query, first specify the database you want to query by clicking on the database name in the Database folder in the Object Browser to select it (alternatively, select it in the database drop-down list). Next, click the **New Query** button in the **Standard toolbar**. A tabbed window will appear along with the SQL Editor toolbar. In the new window, type the text of the SQL query you want to run, and then click the **Execute** button in the SQL Editor toolbar.

The results appear in a tabbed **Results** window below the query window in a spreadsheet style display. The size of the query window and the results window can be adjusted, and the column widths in the results display can be modified using the standard Windows drag-and-drop technique to help make more data visible. You can run multiple queries at the same time—clicking the New Query button again will open another tabbed query window.

1. How do you install the Microsoft SQL Server 2016 ODBC client? Which ODBC software is installed in the process?

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

1. How do you import Microsoft Excel data into an SQL Server table?

The first step is to normalize the data in the Microsoft Excel worksheet into one or more correctly formatted worksheets. The worksheets should not have titles, but should include the proper column headings.

We then use the **SQL Server Import and Export Wizard** as our tool for data import. This tool, however, has some glitches, and we must use some “work arounds.”

1. 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?

Because the **SQL Server Import and Export Wizard** has some glitches, and we must use some “work arounds.” One of these is storing data in a temporary table, creating a properly structured table, and then moving (or copying) the data to the properly structured table. The steps are:

* 1. Import the Microsoft Excel data into a temporary table.
  2. Use an SQL CREATE TABLE statement—written and saved in an SQL Script—to create the needed properly structured table.
  3. Use an SQL bulk INSERT statement—written and saved in an SQL Script—to populate the new table. Bulk INSERT statements use an SQL SELECT statement within the INSERT statement to determine what data will be selected for insertion into the new table.
  4. [Optional] Delete the temporary table.

1. How do you create user accounts in SQL Server 2016? How do you give them appropriate permissions to a specific database?

SQL Server user accounts are created and user permissions are managed in the SQL Server Management Studio.

* 1. In the Object Explorer, right-click the **Security | Logins** folder to display a shortcut menu.
  2. In the shortcut menu, click the **New Login** command to display the **Login – New** dialog box.
  3. In the Login – New dialog box, enter the necessary data for the new user.
  4. To grant permissions, click the **User Mapping** button. Assign specific database permissions on the User Mapping page.
  5. Click the **OK** button to create the new user account.

1. Why would you want to create an ODBC connection to link a Microsoft Access 2016 to an SQL Server 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.

We can connect Microsoft Access 2016 to SQL Server via an **Open Database Connectivity (ODBC)** link between the two.

1. What is an ODBC DSN? Why is one needed?

An ODBC DSN is data source, and DSN stands for data source name. The DSN stores the actual data, such as the specific database to be used, and the user login name and password, needed to establish a ODBC link to a database.

1. How do you create an ODBC connection to link a Microsoft Access 2016 database to an SQL Server Database?

A DSN may be created independently of a specific use by using the Microsoft **ODBC Data Source Administrator** utility provided with the Windows operating system. We describe how to do this in Chapter 7.

However, when connecting a Microsoft Access 2016 database to an SQL Server database, we use the Microsoft Access 2106 **Get External Data – ODBC Database** Wizard. The Wizard includes the steps necessary to create the needed ODBC DSN to connect to the SQL Server database as part of the process. Thus, we create the ODBC connection during the connection process, and no separate DSN creation is needed.

ANSWERS TO EXERCISES

1. 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 2016 Management Studio is correctly installed.

Installation is easy and straightforward. The complete instructions on how to download and install SQL Server 2016 Developer edition are in Appendix A, and will not be repeated here.

1. If you haven’t already done so, work through the steps described in this appendix to create and populate the WP database.

For table creation, use the file: **DBC-e08-MSSQL-WP-Create-Tables.sql**

For data entry, use the file: **DBC-e08-MSSQL-WP-Insert-Data.sql**

1. Using SQL Server 2016 and the Microsoft SQL Server Management Studio, run the 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

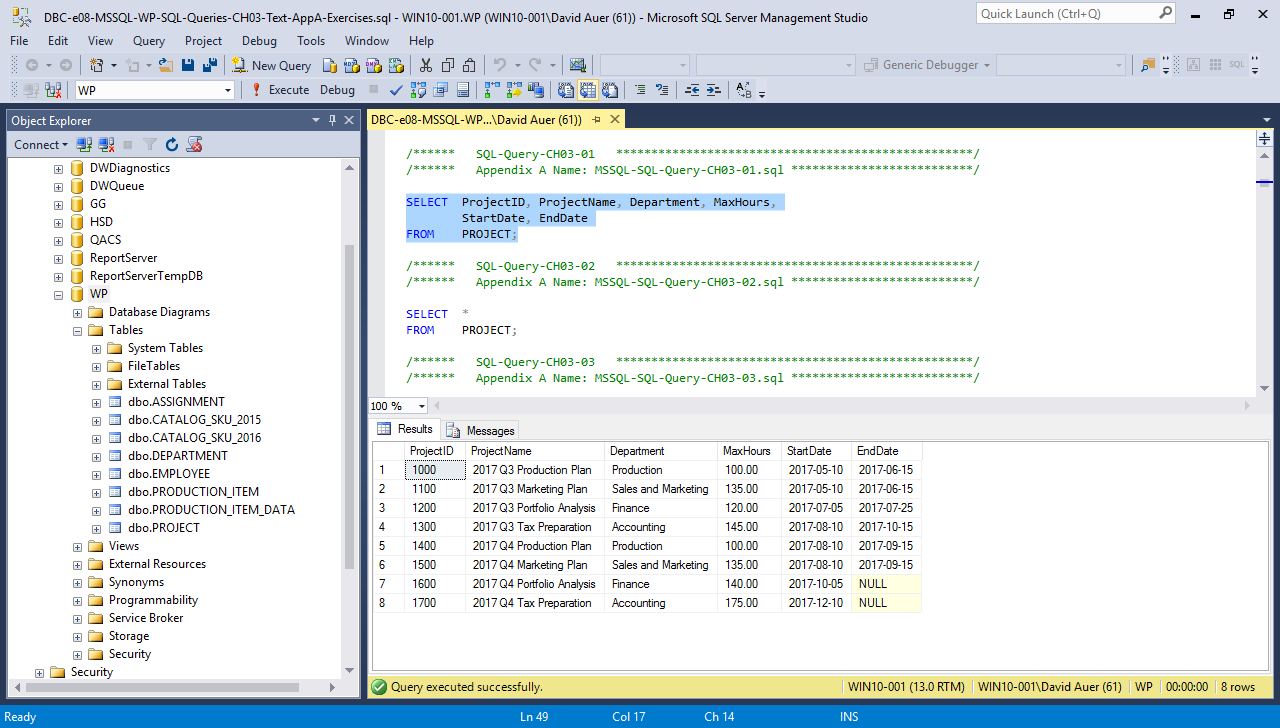
Save each query as follows:

* 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.

The solutions are in the script file:

**DBC-e08-MSSQL-WP-SQL-Queries-CH03-Text-AppA-Exercises.sql**

**DO NOT RUN THIS FILE AS A SCRIPT!** You can run individual queries from the script file by highlighting them and then clicking the Execute button, as shown in the screen shot on the next page.



This is the selected query and only it will be run when you click the **Execute** button

1. Use Microsoft SQL Server Management Studio to run one or more of the saved SQL queries you created in question A.14:

* 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.

These questions are self-explanatory and do not require separate solutions.

1. 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.)

The solution for this Exercise is the same as the solution to Exercise 3.59. See the Instructor's Manual for Chapter 3 and the solutions in the file:

**DBC-e08-MSSQL-WP-SQL Queries-CH03-Exercises.sql**

1. 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 saved query name will be MSSQL-SQL-Query-AWE-3-3-E.sql.

The solution for this Exercise is the same as the solution to Exercise 3.60. See the Instructor's Manual for Chapter 3 and the solutions in the file:

**DBC-e08-MSSQL-WP-SQL Queries-CH03-Exercises.sql**

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

This is self-explanatory. This exercise is intended to make sure that the student is prepared to answer exercise A.31.

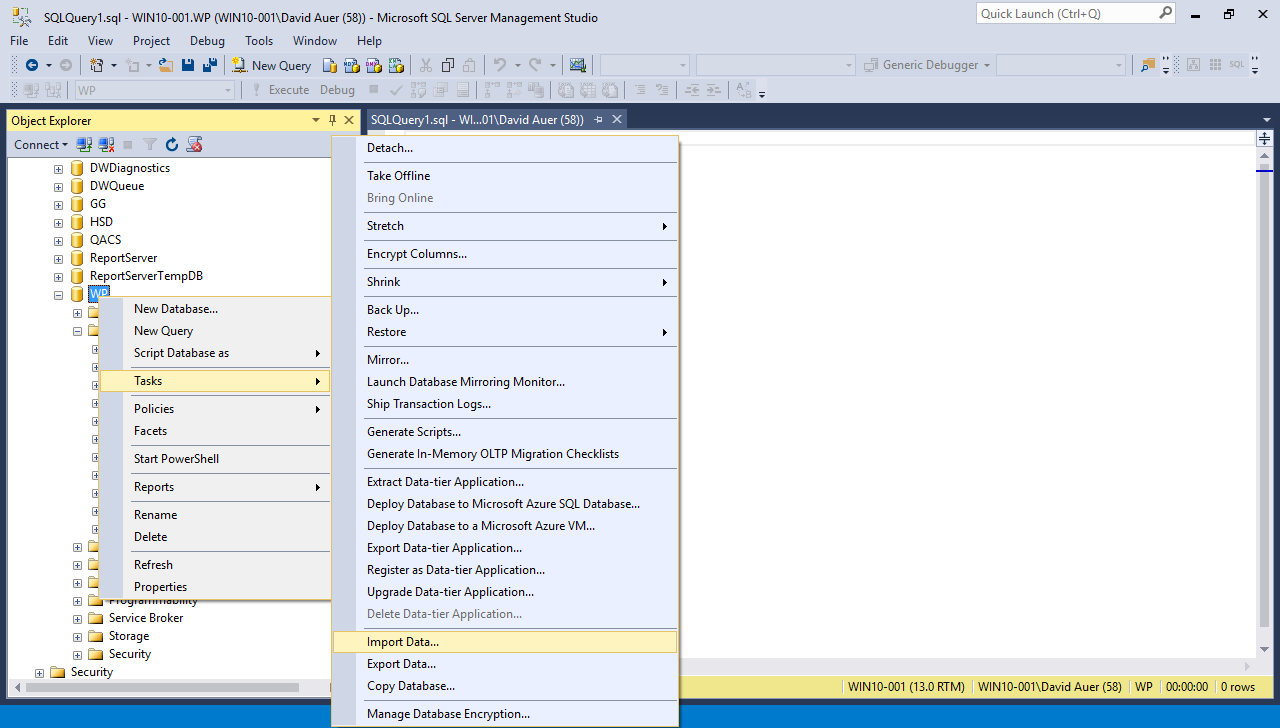
1. Import the COMPUTER\_ASSIGNMENT table from Microsoft Excel in the SQL Server 2016 WP database as explained in the text. How should this table be linked to the EMPLOYEE table and the COMPUTER table by foreign keys? Be sure to include these foreign keys in your final COMPUTER\_ASSIGNMENT table structure when you create it in SQL Server 2016.

To Import the COMPUTER\_ASSIGNMENT table:

1. In the Microsoft SQL Server Management Studio, expand the **WP** database.

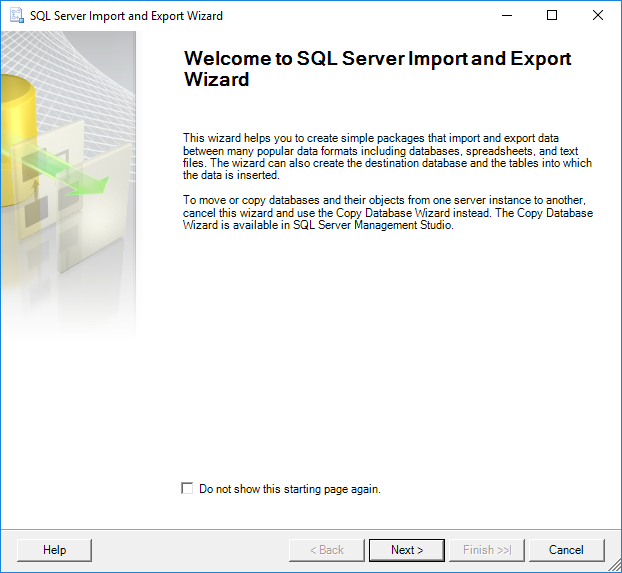
2. Right-click on the WP database object to display a shortcut menu, and in the shortcut menu click on the **Tasks** command to display the Tasks menu.

3. In the Task menu, click the **Import Data** command to launch the SQL Server Import and Export Wizard.



Click the **Import Data** command

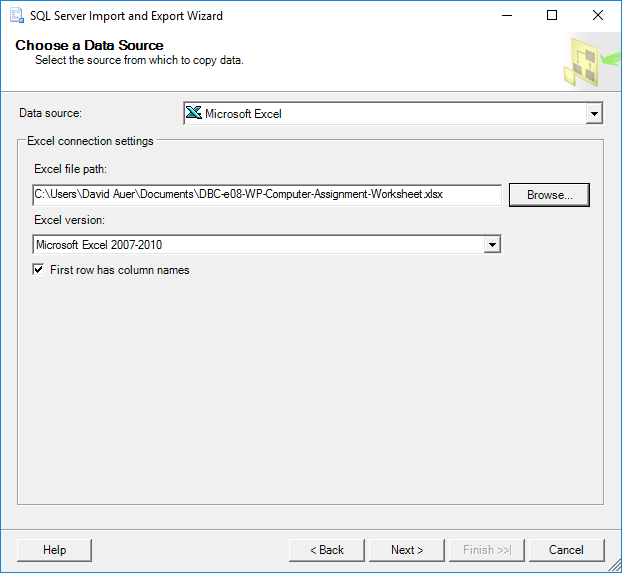
4. On the **Welcome to SQL Server Import and Export Wizard** page, click the **Next** button to display the Choose a Data Source page.



Click the **Next** button

5. On the **Choose a Data Source** page select **Microsoft Excel** as the data source.

6. On the **Choose a Data Source** page, browse to the location of the Microsoft Excel **file**, select **Microsoft Excel 2007-2010** as the Excel version (there is a glitch if Excel 2013 or Excel 2016 is used) from the drop-down list, and make sure the check box for **First row has column names** is checked.



Select **Microsoft Excel 2007-2010**

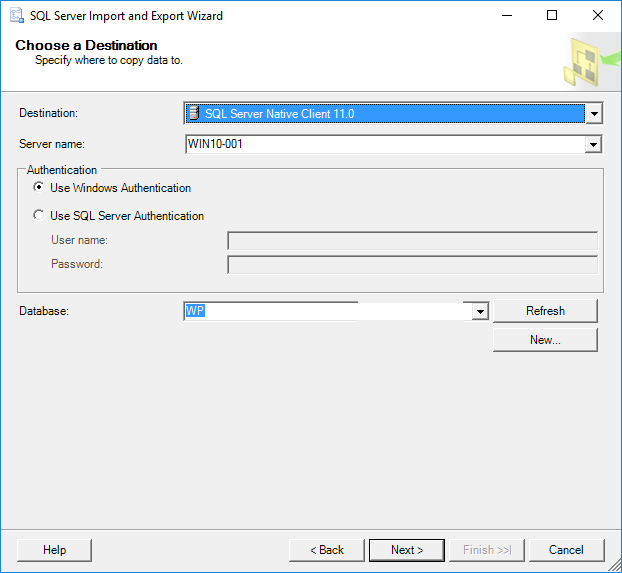
Make sure check box is checked

Click the **Next** button

Browse to the **Excel file**

Select **Microsoft Excel**

7. Click the **Next** button to display the Choose a Destination page, and select SQL Server Native Client 11 as the destination. The WP database values are automatically supplied, and there is nothing to change.



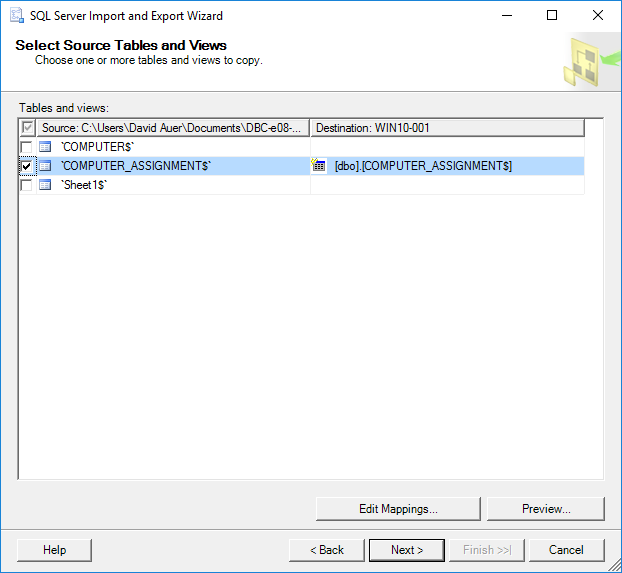
Click the **Next** button

Select **SQL Server Native Client 11.0**

8. Click the **Next** button to display the **Specify Table Copy or Query** page. The defaults are correct here.

9. Click the **Next** button to display the **Select Source Tables and Views** page, and check the **‘COMPUTER\_ASSIGNMENT$’** check box in the Source column. The table name  **[dbo].[COMPUTER\_ASSIGNMENT$]** is generated and displayed in the Destination column. This is the name we will use for the temporary table in the WP database.

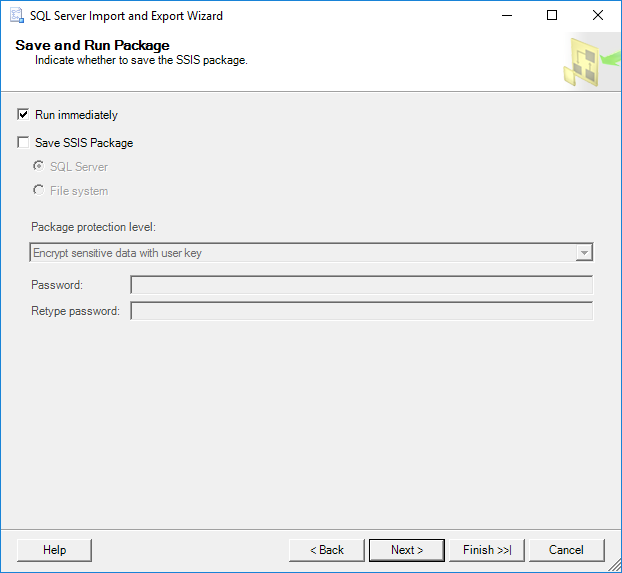
Select **SQL Server Native Client 11.0**



Select **‘COMPUTER\_ASSIGNMENT$’**

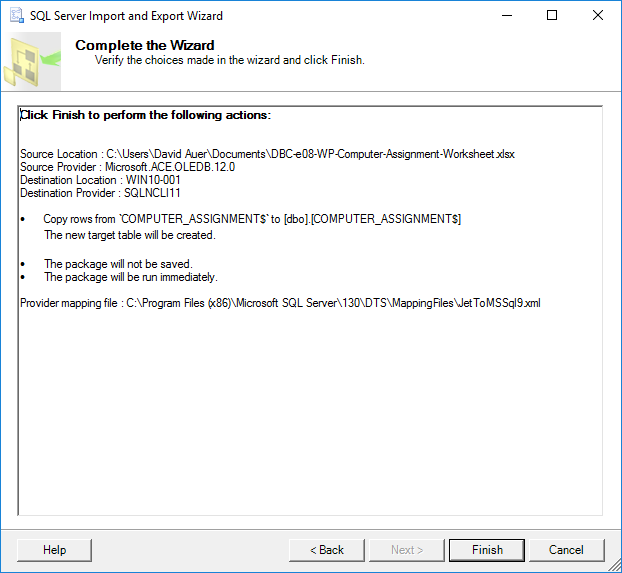
Click the **Next** button

10. Click the **Next** button to display the **Save and Run Package** page. The defaults are correct.



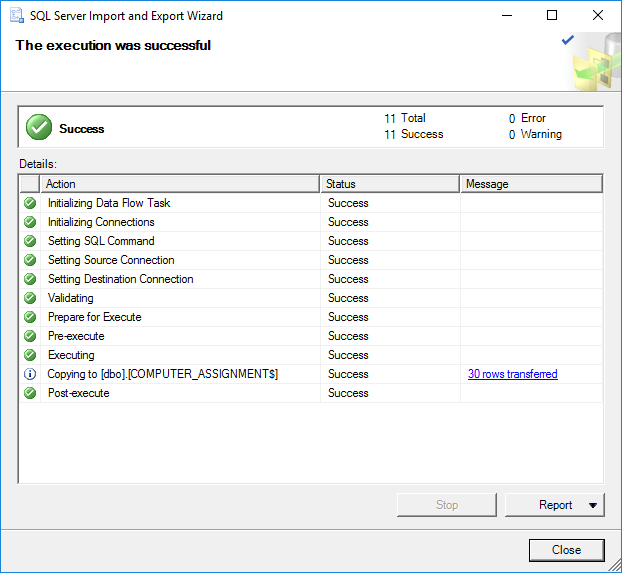
Click the **Next** button

11. Click the **Next** button to display the **Complete the Wizard** page. This is a summary page. Click the **Finish** button.



Click the **Finish** button

12. The SQL Server Import and Export Wizard runs the actual import, and then displays the **The execution was successful** page. Note that there are no errors in the process. Click the **Close** button to close the Wizard.



Click the **Close** button

Now we created the actual COMPUTER\_ASSIGNMENT table and copy the data to it.

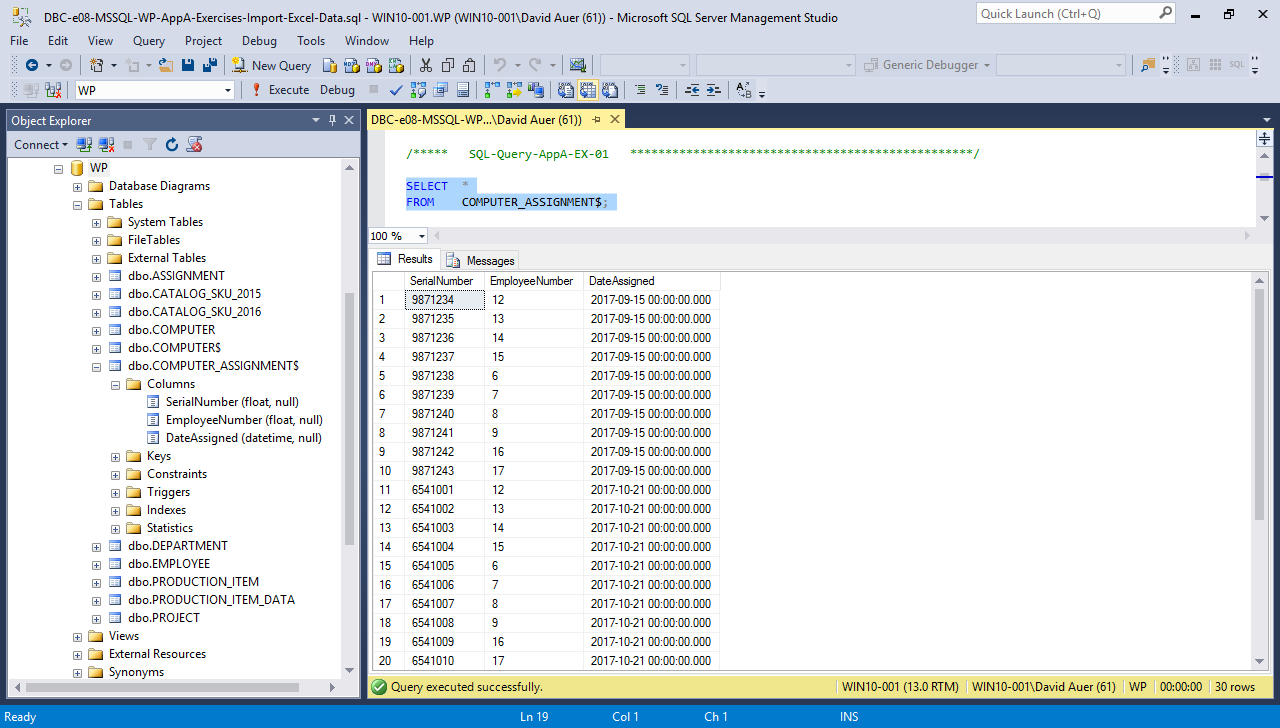
1. In SQL Server Management Studio, refresh the **WP** database. In Object Explorer, expand the WP database, then expand the Tables object, then expand the **dbo.COMPUTER\_ASSIGNMENT$** object, and finally expand the Columns object.

2. Open a New Query window, and run SQL-Query-AppA-EX-01:

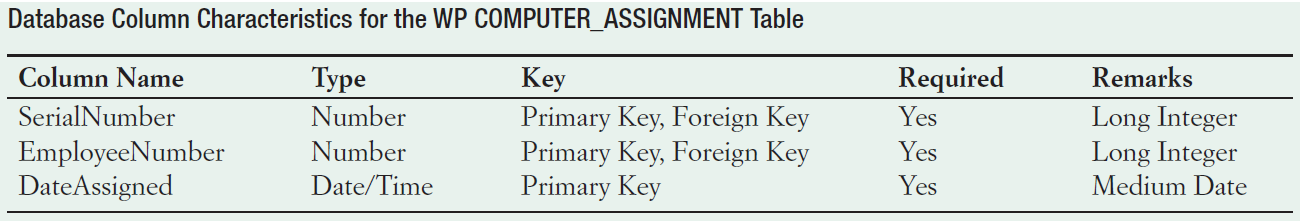
/\* \*\*\* SQL-Query-AppA-EX-01 \*\*\* \*/

SELECT \*

FROM COMPUTER\_ASSIGNMENT$;



3. Now we have to create the final COMPUTER\_ASSIGNMENT table in the WP database. In the Microsoft SQL Server Management Studio, write the SQL CREATE TABLE statement for the COMPUTER\_ASSIGNMENT table based on the column characteristics in Figure A-65 (these are Microsoft Access 2016 specifications).



**Figure A-65 — Database Column Characteristics for the WP COMPUTER\_ASSIGNMENT Table**

One important consideration here is foreign key constraints. Note that we link to both the EMPLOYEE table using EmployeeNumber and to the COMPUTER table using SerialNumber. What referential integrity constraints should we use?

For EMPLOYEE, EmployeeNumber is a surrogate key and is never updated. Further, WP never drops employee records because of record keeping requirements. Therefore, we will never cascade updates or deletes for this primary key.

FOR COMPUTER, SerialNumber never changes. However, when we remove a computer from the WP computer inventory, we do not need to keep historical records of having had that computer. Therefore, while we do not cascade updates, we will cascade deletions for this primary key.

Here is the final SQL statement to create the COMPUTER\_ASSIGNMENT table:

/\* \*\*\* SQL-CREATE-TABLE-AppA-EX-01 \*\*\* \*/

CREATE TABLE COMPUTER\_ASSIGNMENT(

SerialNumber Int NOT NULL,

EmployeeNumber Int NOT NULL,

DateAssigned Date NOT NULL,

CONSTRAINT COMP\_ASSIGN\_PK PRIMARY KEY

(SerialNumber, EmployeeNumber, DateAssigned),

CONSTRAINT C\_A\_EMP\_FK FOREIGN KEY(EmployeeNumber)

REFERENCES EMPLOYEE(EmployeeNumber)

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT C\_A\_COMP\_FK FOREIGN KEY(SerialNumber)

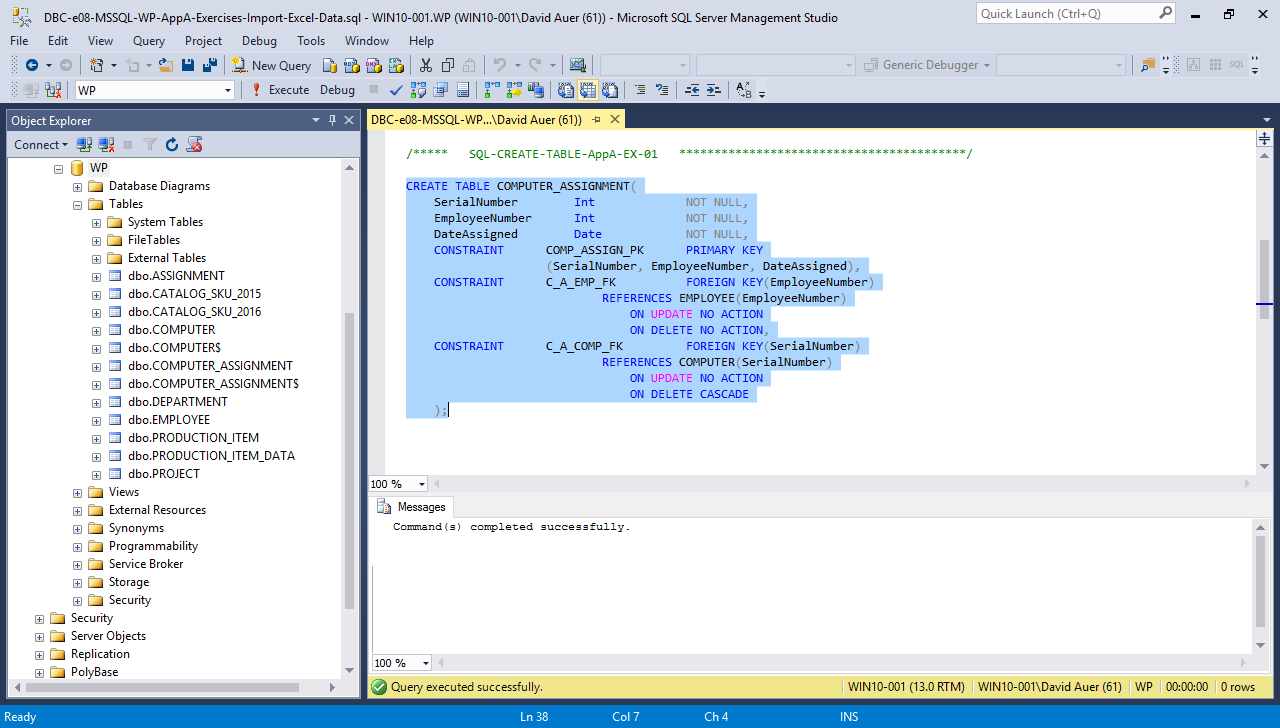
REFERENCES COMPUTER(SerialNumber)

ON UPDATE NO ACTION

ON DELETE CASCADE

);

4. Run the SQL-CREATE-TABLE-AppA-EX-01 statement. The result is shown in the screen shot on the next page.



5. To copy the imported data from the temporary COMPUTER\_ASSIGNMENT$ table to the final COMPUTER\_ASSIGNMENT table, use the SQL bulk INSERT statement   
SQL-INSERT-AppA-EX-01. Note that we use an ORDER BY clause in the SELECT statement to order the data inserted into the COMPUTER\_ASSIGNMENT table:

/\* \*\*\* SQL-INSERT-AppA-EX-01 \*\*\* \*/

INSERT INTO dbo.COMPUTER\_ASSIGNMENT

(SerialNumber, EmployeeNumber, DateAssigned)

SELECT SerialNumber, EmployeeNumber, DateAssigned

FROM COMPUTER\_ASSIGNMENT$

ORDER BY SerialNumber, EmployeeNumber, DateAssigned;

6. After running the SQL-INSERT-AppA-EX-01 statement, run SQL-Query-AppA-EX-02. The result is shown in the screen shot on the next page.

/\* \*\*\* SQL-Query-AppA-EX-02 \*\*\* \*/

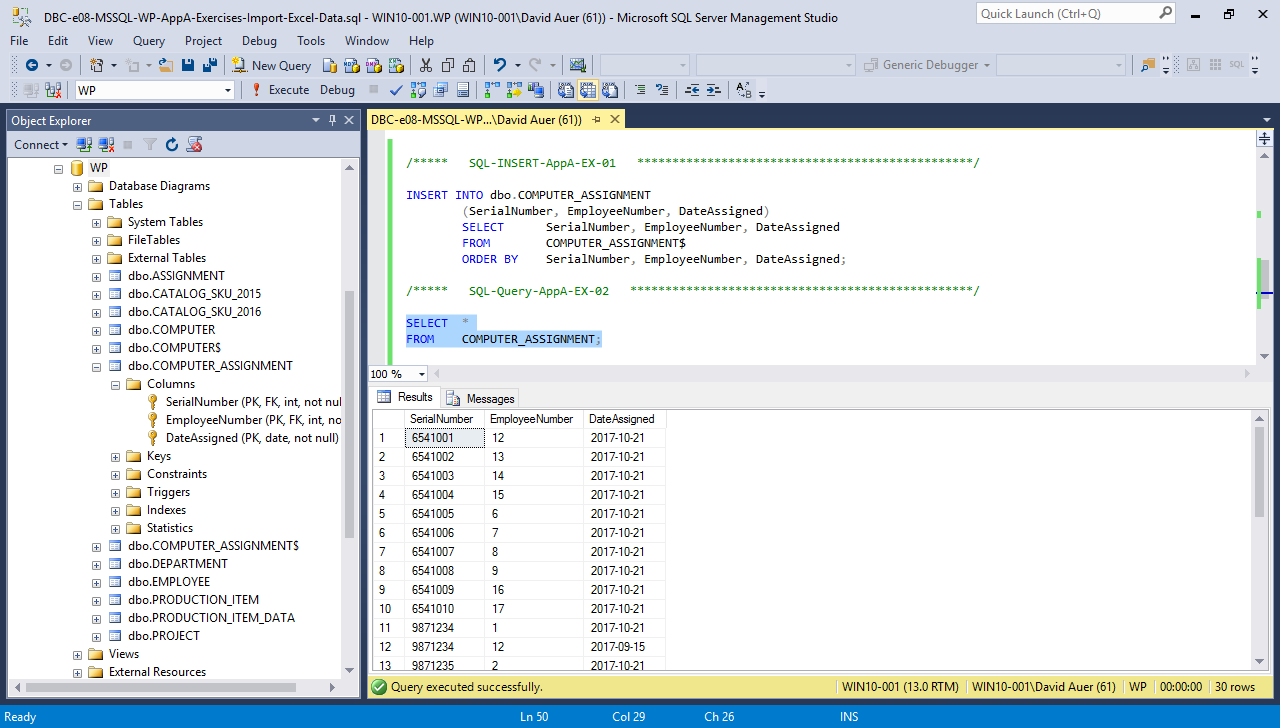
SELECT \*

FROM COMPUTER\_ASSIGNMENT;

7. Dropping the temporary COMPUTER\_ASSIGNMENT$ table is optional—if you drop it, be sure you drop the right table!

/\* \*\*\* SQL-DROP-TABLE-AppA-EX-01 \*\*\* \*/

DROP TABLE COMPUTER\_ASSIGNMENT$;



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

1. 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.

This is self-explanatory. This exercise is intended to make sure that the student is prepared to answer exercise A.33.

1. 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.

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.

3. The **General** page of the Login - New dialog box is displayed.

4. In the Login - New dialog box General page, type in the login name **WP-Reader**.

5. Click the **SQL Server authentication** radio button.

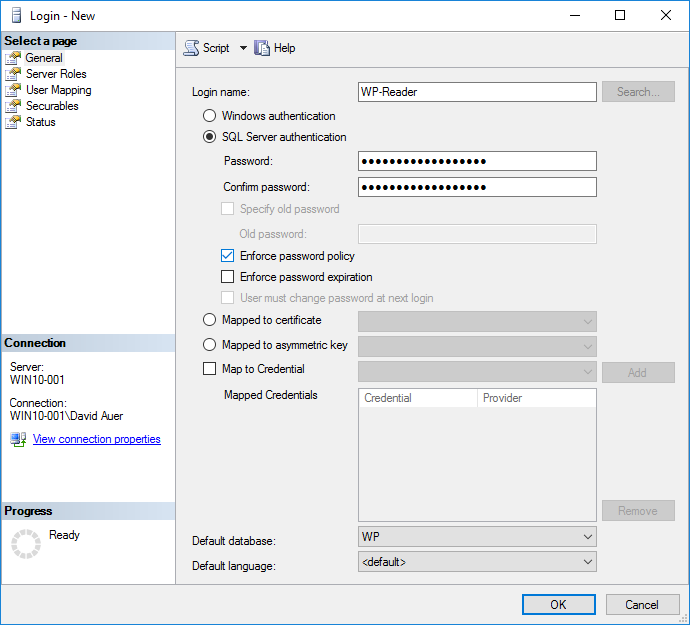
6. Uncheck the **User must change password at next login** password setting.

7. Uncheck the **Enforce password expiration** password setting.

8. Type in the password **WP-Reader+password** in both the Password and Confirm password text boxes.

9. Select **WP** as the Default database from the Default database drop-down list.

Enter login name   
**WP-Reader**



Select **WP** from drop-down list

Uncheck   
check box

Uncheck   
check box

Confirm password

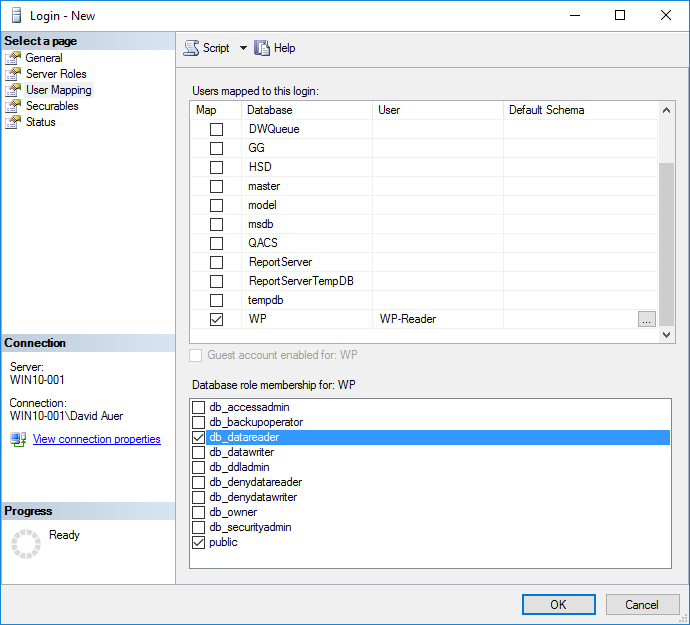
Select **SQL Server authentication**

Enter password

10. Double check your settings, and then click the **User Mapping** button shown to display the User Mapping page of the Login – New dialog box.

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.

12. In the **Database role memberships for: WP** list, leave already checked the public role selected and additionally check the check box for the db\_datareader database roles. These permissions give WP-User the necessary rights to read data from the WP database tables, but do not grant permission to write new or revised data to the WP database tables.



Click the **OK**  
button

Check the **db\_datareader** check box

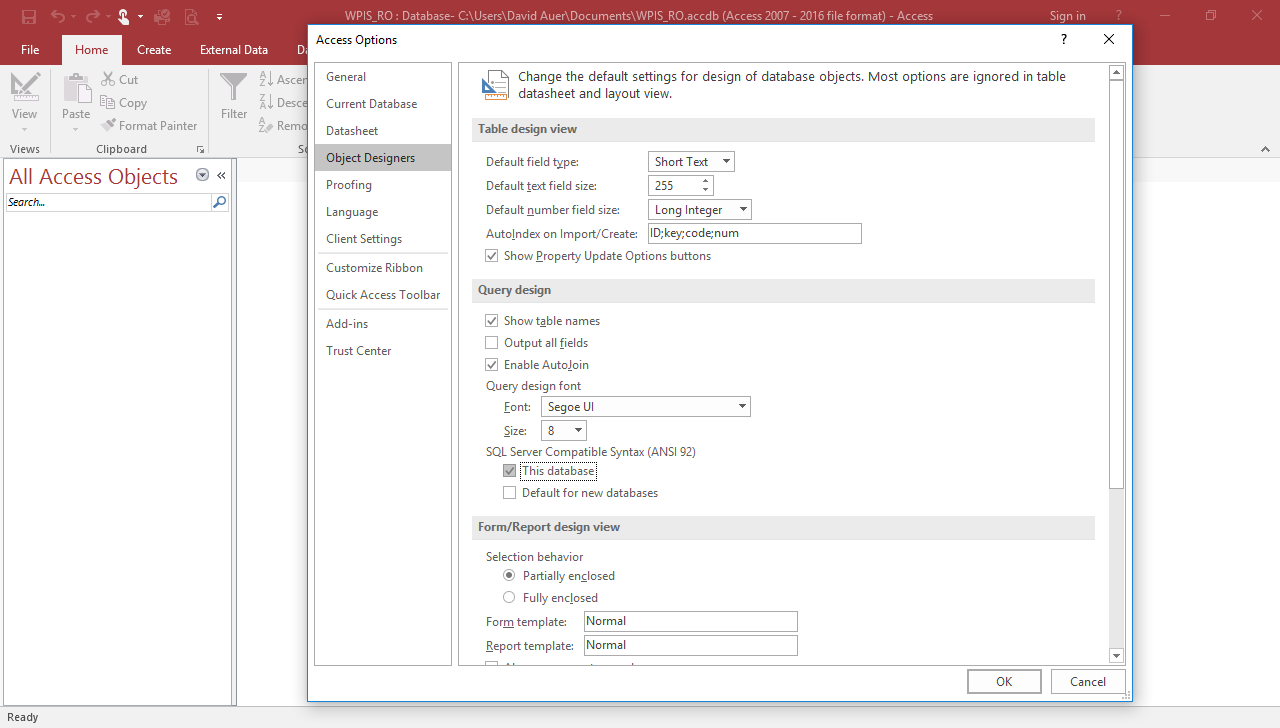
Check the **WP**  
check box

The **User Mapping** page

13. Double check your settings, and then click the **OK** button.

14. The new user login WP-Reader is created, and assigned the specified set of permissions to the WP database.

1. 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)****.*



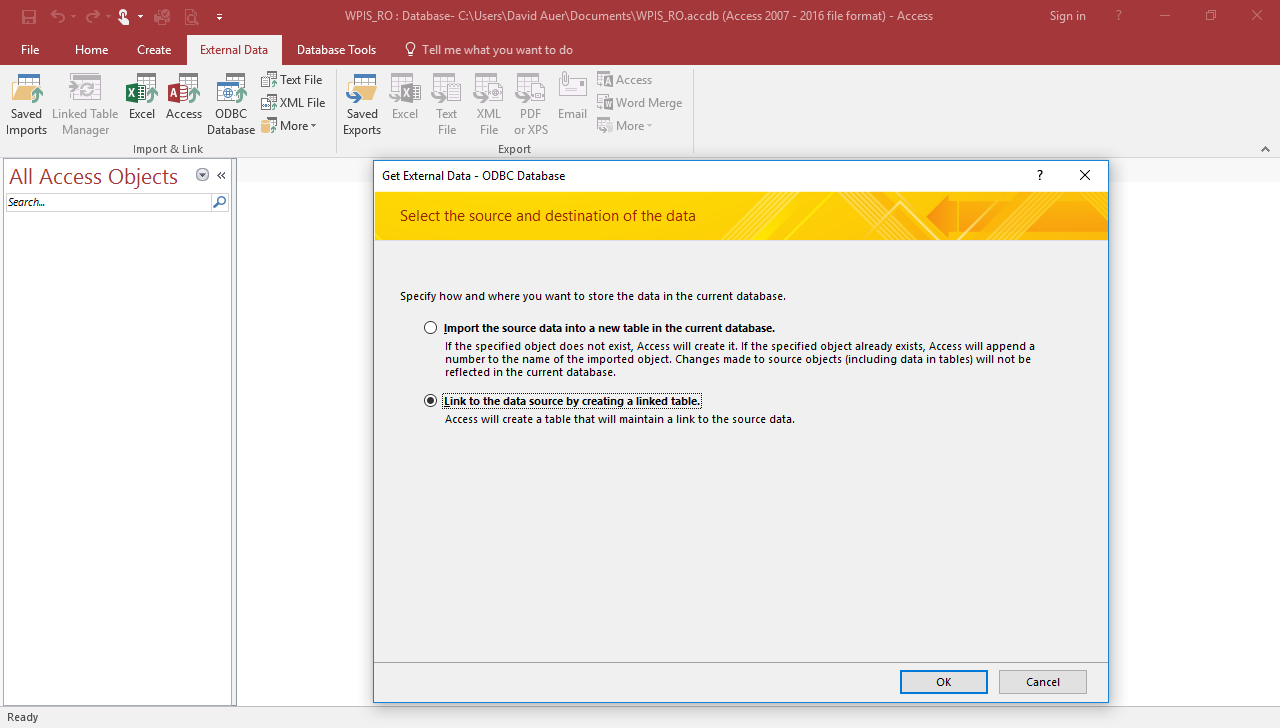
Select the **This database** radio button

* 1. *Link the WPIS\_RO.accdb database to the SQL Server 2016 WP database. When you create your File Data Source DSN, name the DSN* ***WPRO****, and use the WP-Reader user account (as detailed in Exercise A.33) for SQL Server authentication.*

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.

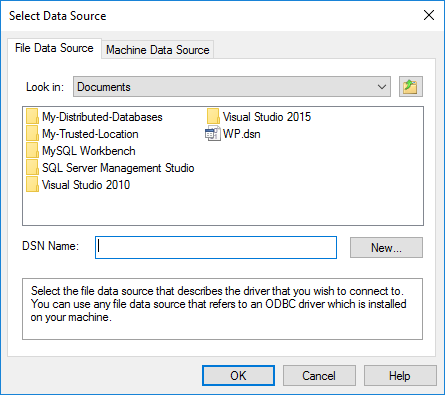
3. In the Get External Data – ODBC Database Wizard dialog box, the **Select the source and destination of the data** page is displayed. Click the **Link to the data source by creating a linked table** radio button, as shown in the screen shot on the next page.



Select the **Link to the data source by creating a linked table** radio button

4. Click the **OK** button.

5. The **Select Data Source** dialog box is displayed. This is the dialog box that we will use to create the needed OBDC DSN. In the Select Data Source dialog box, make sure the **File Data Source** tab is selected.



The **OK** button

The **Select Data Source** dialog box

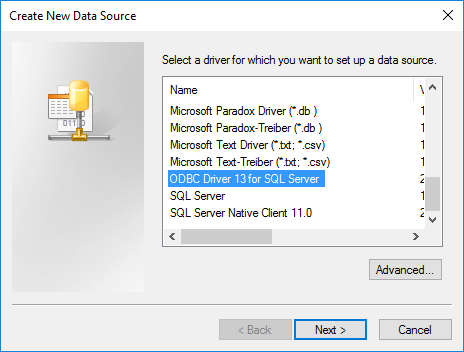
The **DSN Name** text box

The **New** button

The **File Data Source** tab

6. Click the **New** button to display the **Create New Data Source**.

7. 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.



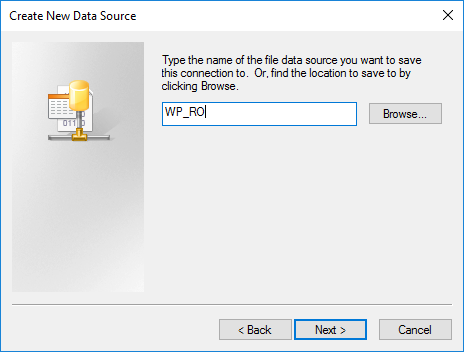
Select the **OBDC Driver 13 for SQL Server** driver

The **Next** button

The **Create New Data Source** dialog box

8. Click the **Next** button.

9. The next page of the Create New Data Source dialog box provides a text box for naming the new DSN. Type in **WP\_RO**.

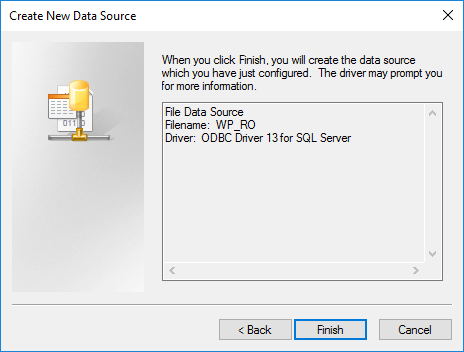


Type in the name **WP\_RO**

The **Next** button

10. Click the **Next** button.

11. 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.



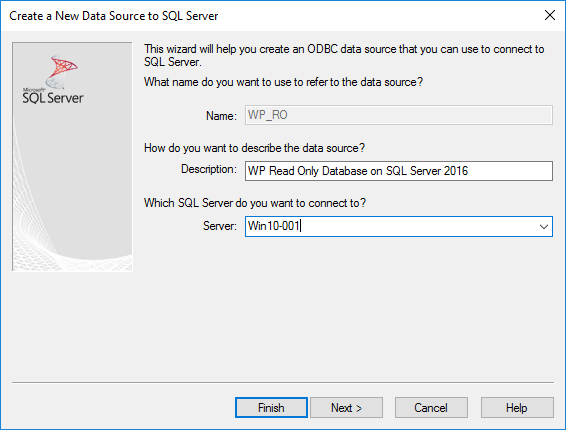
The **Finish** button

10. Click the **Finish** button.

11. The **Create a New Data Source to SQL Server** Wizard dialog box is displayed, and the DSN name of WP is already assigned.

12. In the **Description** text box enter the text **WP Read Only Database on SQL Server 2016**.

13. 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. The dialog box now appears as shown here.



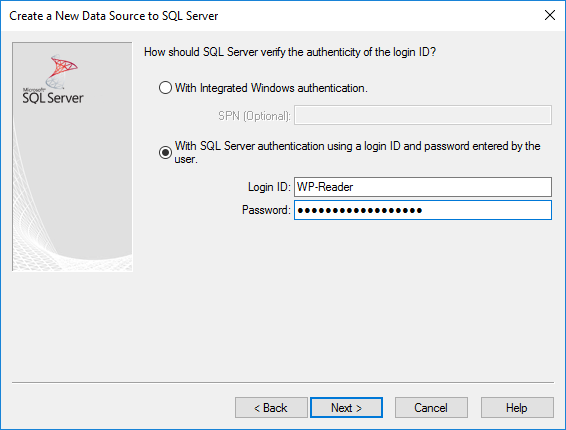
The **Next** button

Type in the **SQL Server** name

Type in the description

The DSN **WP\_RO**

14. 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-Reader** and the password **WP-Reader+password**.



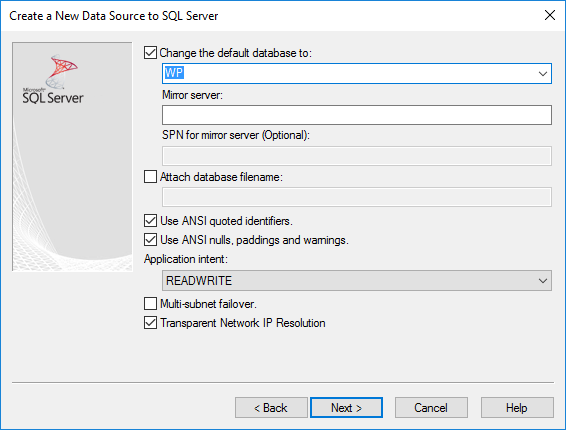
Type in the password   
**WP-Reader+password**

Type in the login ID   
**WP-Reader**

Select this radio button

The **Next** button

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.



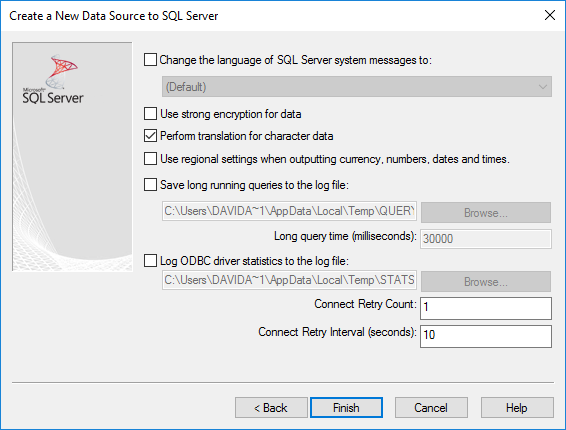
Type in the password   
**WP-Reader+password**

Type in the login ID   
**WP-Reader**

Select this radio button

The **Next** button

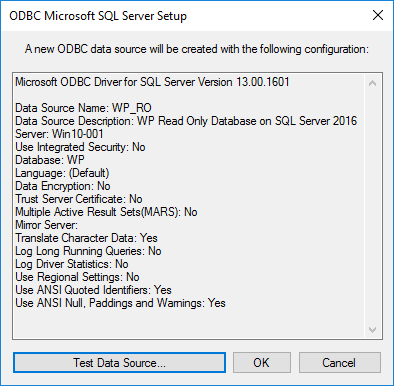
17. Click the **Next** button. An additional set of database settings page is displayed. The defaults shown here are correct.



The **Finish** button

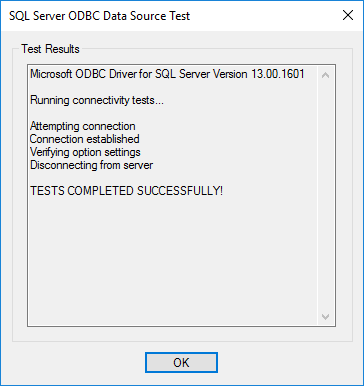
18. Click the **Finish** button. The **ODBC Microsoft SQL Server Setup** dialog box is displayed, showing the settings that will be used to create the DSN. Additionally, the dialog box has a Test Data Source button that can be used to test the DSN before it is created.

19. Click the **Test Data Source** button.



The **Test Data Source** button

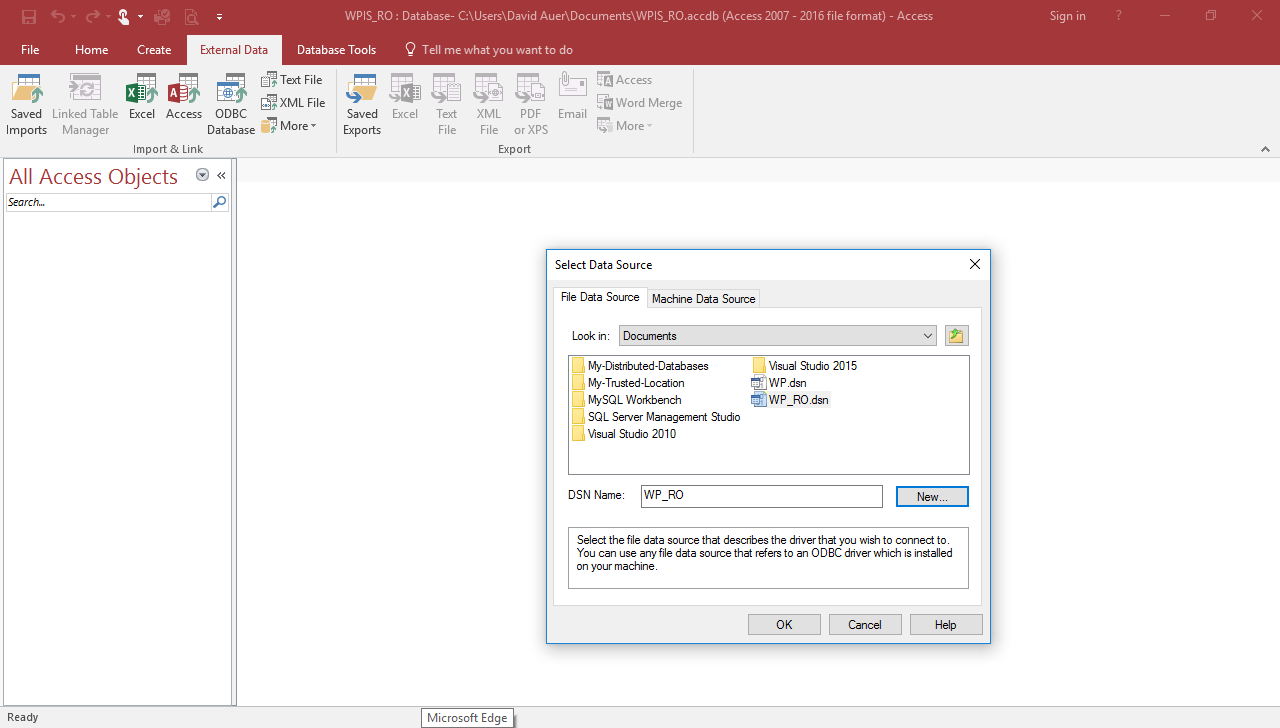
20. If all the settings are correct, the **SQL Server ODBC Data Source Test** dialog box is displayed with the message “TESTS COMPLETED SUCCESSFULLY”. In the SQL Server ODBC Data Source Test dialog box, click the **OK** button.



The **OK** button

21. The **ODBC Microsoft SQL Server Setup** dialog box is displayed. Click the **OK** button.

22. The **WP\_RO DSN** file data source is created and displayed in the **Select Data Source**. Now that the DSN is completed, we can finish linking the Microsoft Access 2016 database to the SQL Server database. In the Select Data Source dialog box, click the **OK** button.



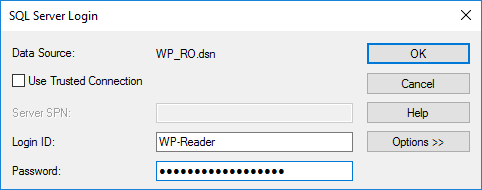
The **WP\_RO** DSN

The **OK** button

* 1. *Import all existing tables (including the COMPUTER and COMPUTER\_AUTHENTICATION tables if they have been imported as detailed in Exercises A.30 and A.31).*

**Note that the steps in this question are a continuation of the steps in question 2—the Microsoft Access 2016 Get External Data – ODBC Database Wizard continues to run after we close the Select Data Source dialog box!**

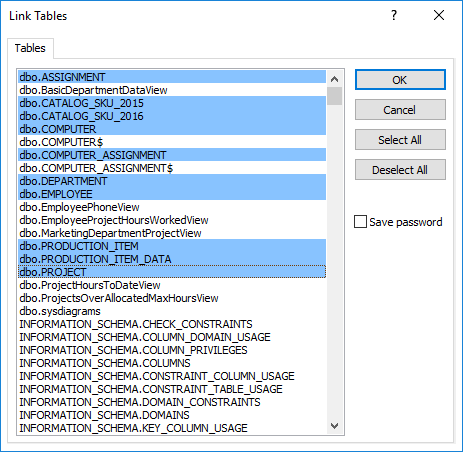
1. The **SQL Server Login** dialog box is displayed. Enter the password   
**WP-Reader+password** in the Password text box, and then click the **OK** button.



Enter the password   
**WP-Reader+password**

The **OK** button

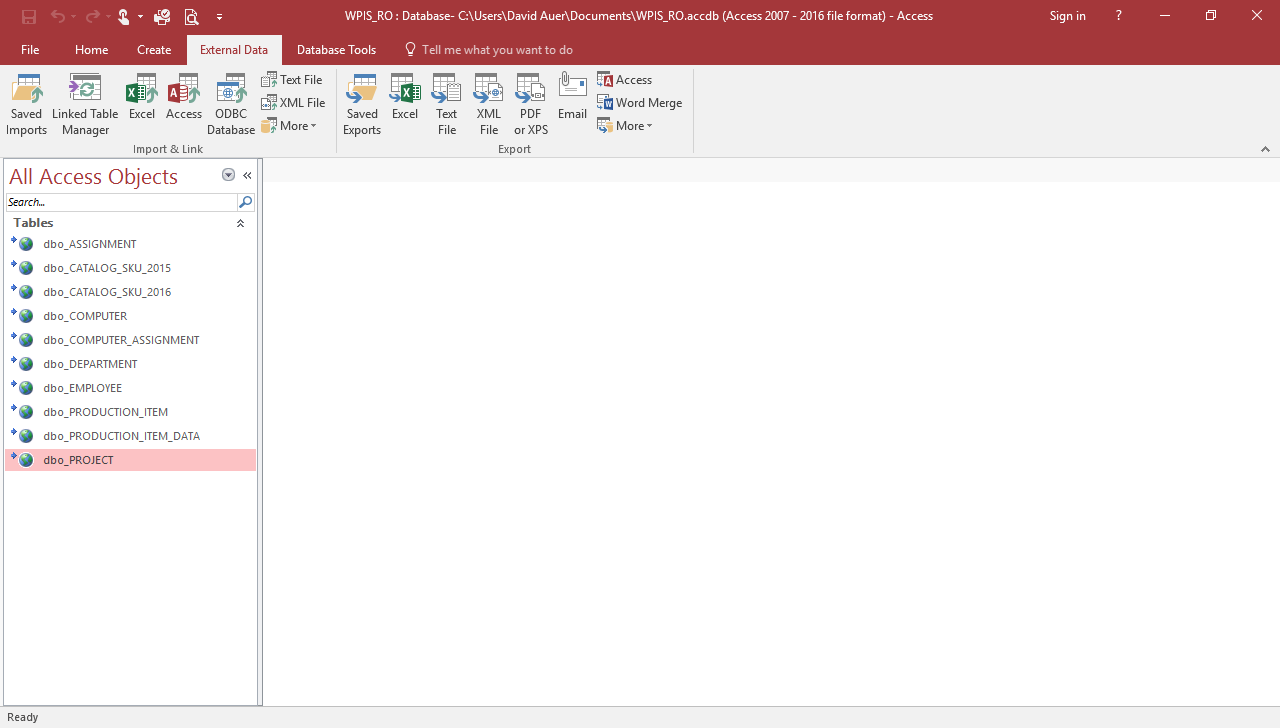
2. The **Link Tables** dialog box is displayed. Select the ten tables in the WP database as shown. To select the **dbo.ASSIGNMENT** table, click on it. To add each additional table to the selection, **press the Ctrl key and then click on the table name**. ***dbo*** stands for database owner and is commonly shown in SQL Server object names (technically, it is a schema name).



Select these tables

The **OK** button

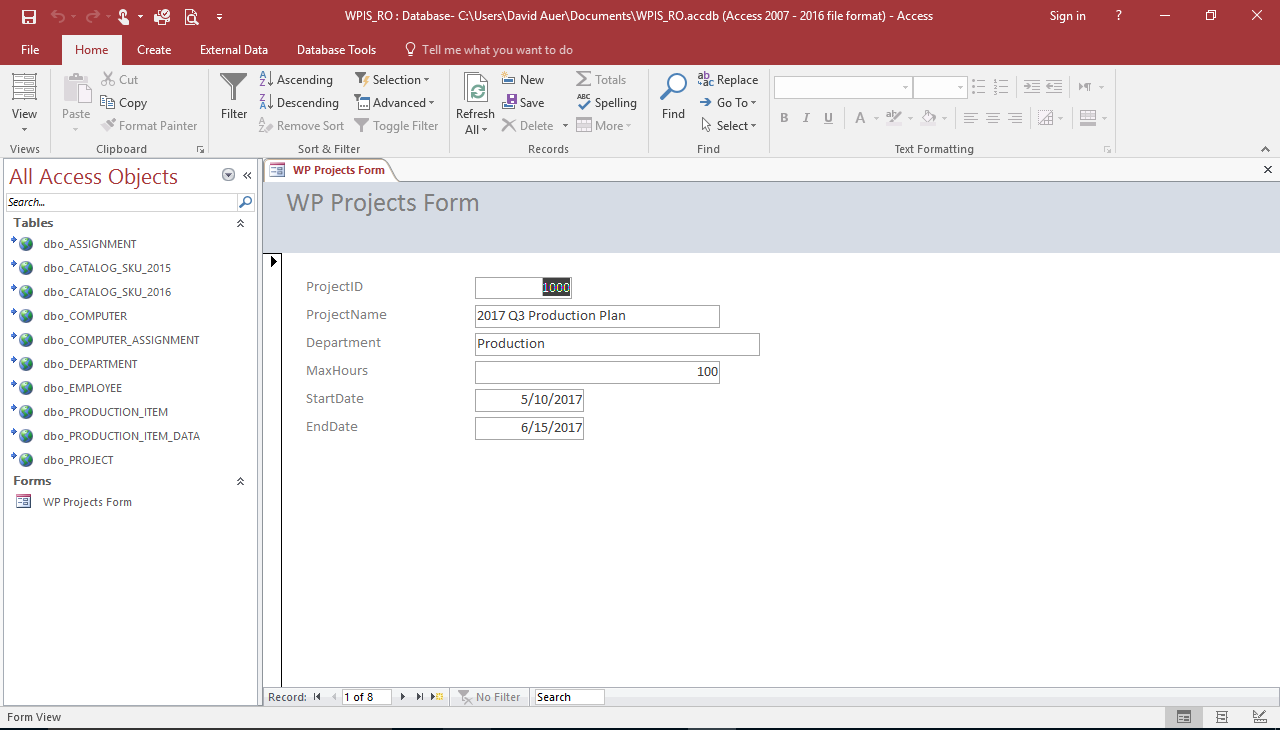
3. Click the **OK** button. The ODBC links between the WP\_RO.accdb Microsoft Access 2016 and the WP database in SQL Server 2016 are completed.



The linked SQL Server 2016 tables

* 1. *Create a form to show all the data in PROJECT table named* ***WP Projects Form****.*

This is done using Microsoft Access 2016 techniques as discussed the Chapter 1 section of “The Access Workbench.”



* 1. *Create a report to show all the data in the PROJECT table named* ***WP Projects Report****.*

This is done using Microsoft Access 2016 techniques as discussed in the Chapter 4 section of “The Access Workbench.”

