Database Concepts

8th Edition

David M. Kroenke • David J. Auer

Scott L. Vandenberg • Robert C. Yoder

Online Appendix B

Getting Started with Oracle Database XE

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 Houpt
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.

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 B — 10 9 8 7 6 5 4 3 2 1

Appendix Objectives

- Learn how to install Oracle Database XE
- Learn how to install Oracle SQL Developer
- Learn how to create a database in Oracle Database XE
- 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 Oracle ODBC Client software
- Learn how to use Microsoft Access as a front end to an Oracle Database XE database
- 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 Oracle Database XE.
- Describe how to install and use Oracle Database XE and the Oracle SQL Developer GUI.
- Describe how to use Microsoft Access as a frontend to an Oracle Database XE database.
- Describe and use DBMS data import techniques to import Microsoft Excel worksheet data into an Oracle database.

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

- Before starting Chapter 3 on SQL, install Oracle Database XE and Oracle SQL Developer by working through the appendix up to and including "How Do I Install Oracle SQL Developer?"
- 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 Oracle Database XE?" Both Chapter 3 and this appendix use the same WP database, so you can run the SQL Statements shown in Chapter 3 yourself and see the results.
- If you are interested in using Microsoft Access as a frontend to an Oracle Database XE database, then work through the section "How Do I Create an ODBC Connection from Microsoft Access 2016 to an Oracle Database XE Database?"
- Finally, work through the section "How Do I Import Microsoft Excel Data into an Oracle Database Table?" in this appendix to understand how to import Microsoft Excel data in a database table.

What Is Oracle Database XE?

Oracle Corporation's **Oracle Database 12*c*** is an enterprise-class DBMS that has been around for many years. It is important to distinguish between the name of the company, *Oracle* (or more completely *Oracle Corporation*), and the name of the product, *Oracle Database*. In earlier years, the Oracle Database DBMS was the main product produced by Oracle, and the word *Oracle* referred to both the company and the DBMS. Thus, in 2001, Oracle 9*i* was released, followed by Oracle 10*g* (with *g* for *grid*, a reference to grid computing) and Oracle Database 11*g*. As this book goes to print, the current version of Oracle Database is Oracle Database 12*c* (with *c* for *cloud computing*), Release 1.

As Oracle Corporation acquired more product lines, however, there was a need to specifically refer to the DBMS in the product name, and the DBMS was renamed *Oracle Database*. Oracle Database 11*g* became available in 2007 as Oracle Database 11*g* and later as Oracle Database 11*g* Release 2. Oracle Database 10*g* (10.2) and 11*g* (11.1) are no longer generally available. Oracle Database 12*c* is available in several versions, which can be reviewed at the Oracle Database 12*c* Release 1 Web site.[1] For our purposes, there are four editions we need to be aware of:

- **Enterprise Edition.** The most powerful and feature-laden version of Oracle Database 12*c*. This version handles as many CPUs and memory as the computer's operating system will support. It has full Data Warehouse Builder features. In addition, OLAP and data mining (Oracle Advanced Analytics) options are available.

- **Standard Edition 2.** This is the basic commercial version of Oracle Database 12*c* Release 1. It does not have the complete feature set of the Enterprise Edition. In particular, it lacks many of

[1] See https://docs.oracle.com/database/121/DBLIC/editions.htm (accessed December 2016).

the high-availability and high-performance options, can utilize only up to 16 CPUs, and includes only limited data warehouse capabilities.

- **Express Edition 11*g* Release 2 (Oracle Database XE).** This is a free downloadable version based on Oracle Database 11*g* Release 2, and Oracle documentation refers to it as **Oracle Database XE**. It has limited features: It supports only one CPU, it has 1 GB of memory, and the maximum database size is only 11 GB. Despite its limitations, it is a great learning tool.

- **Personal Edition.** The Personal Edition 12*c* is available only for Windows and Linux systems. It is intended for single-user use by a developer. It has nearly all the features of the Enterprise Edition.

The Oracle Database XE was introduced with Oracle Database 10*g*, and the current **Oracle Database Express Edition 11g Release 2**, like the SQL Server Express Editions, seems to be designed to compete with MySQL Community Edition (see Appendix C). Although MySQL Community Edition does not have as many features as Oracle Database 12*c* or SQL Server 2016, it is an open source database that has had the advantage of being widely available for download over the Internet. It has become widely used and very popular as a DBMS, supporting Web sites running the Apache Web server. MySQL is now owned by Oracle Corporation.

Note that the full name of the DBMS is Oracle Database Express Edition 11g Release 2, but we will follow the lead of the Oracle documentation and refer to it as Oracle Database XE in this appendix, with the understanding that the term always refers to the most recent released version of the software. We will be working with Oracle Database XE in this appendix, but nearly all of our discussion, aside from installation, is relevant to other Oracle Database versions (12c, 11g Release 2, etc.). Note, however, that you will still hear the Oracle DBMS product referred to as just *Oracle, Oracle 11g Release 2*, *Oracle XE*, *Oracle 12*c, or other similar variations.

Be aware that Oracle Database 12*c* and Oracle Database XE are enterprise-class DBMS products and, as such, are much more complex than Microsoft Access. Further, the basic DBMS product does not include application development tools, such as form and report generators. Regardless of which version of Oracle Database you are going to use, you should install it now.

By The Way

To install Oracle Database XE, 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, right-click the **Start** button in the lower-left corner, then select **System**. Look at the value of **System Type** in the **System** information group.

Why Should I Learn to Use Oracle Database XE?

For the purposes of this book, the most important reason to learn to use Oracle Database XE is that Oracle really handles SQL well. All the SQL commands and keywords in Chapter 3 and Appendix E marked "Does Not Work with Microsoft Access ANSI-89 SQL" will work with Oracle. In addition, Oracle is a widely used enterprise-class DBMS that may be encountered in many places. As such, it provides a very complete set of SQL and other functionality.

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.

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 Oracle Database topics not covered here, including views, stored procedures, triggers, backups and restores, and database security. Views, stored procedures, and triggers are covered briefly in Appendix E. All these topics are covered in more depth in David M. Kroenke and David J. Auer, *Database Processing: Fundamentals, Design, and Implementation*, 14th edition (Upper Saddle River, NJ: Prentice Hall, 2016).

Installing a Loopback Adapter

IMPORTANT NOTE: While the Oracle Database XE does not require an operating system loopback adaptor, the standard version of Oracle Database 12c definitely does. If, for some reason, you are using this book with a non-Express Edition of Oracle Database, you will need to install an operating system loopback adaptor before installing Oracle Database.

If your computer is set up to get an IP number configuration from a **dynamic host configuration protocol (DHCP)** server (which is typical of high-speed Internet connections), then you must install a **loopback adapter**, which assigns a local and fixed IP address to your computer, before installing Oracle Database. It is very important that the loopback adapter be correctly installed *before* the installation of Oracle Database, or the Oracle Database installation will have serious problems.

If you are using a Windows OS version of Oracle Database, see the Microsoft operating system instructions for installing a loopback adapter at the Oracle Web site.[2]

[2] See https://docs.oracle.com/database/121/NTDBI/network.htm (accessed December 2016).

How Do I Install Oracle Database XE?

Although built on a slightly older version of Oracle Database, Oracle Database XE will provide you with all the functionality you need to do the work in this book, and everything you learn will work in Oracle Database 12*c*. The Oracle Database XE installation includes a utility allowing for database administration, which we will use, but for database development (creating and querying tables) we will use Oracle's SQL Developer, which is a separate piece of software. The following sections will describe the installation process for Oracle Database XE and Oracle SQL Developer. Note that you will need administrative privileges on your Windows computer in order to complete the installation.

All the necessary software is freely available for download from Oracle's Web site (www.oracle.com), but the downloads require you to create a (free) account at the Oracle Web site. In the remainder of this appendix, we will be using Windows 10 64-bit versions of all software, but most of the instructions should work for 32-bit operating systems as well, with the exception of the SQL Developer installation: Those differences will be noted below.

To obtain your free Oracle account (if you do not already have one), go to the Oracle home page (www.oracle.com), as shown in Figure B-1(a), and click on the **Sign In/Register** link. In the window that appears, click on the **Create Account** button (Figure B-1(b)) and follow the instructions to create your account. Once your account has been created, you can then revisit the Oracle home page and use the **Sign In/Register** link to login to your account, which will enable you to perform downloads. Alternatively, you can proceed to the download pages without logging in, and when you attempt to download, you will be prompted for a login.

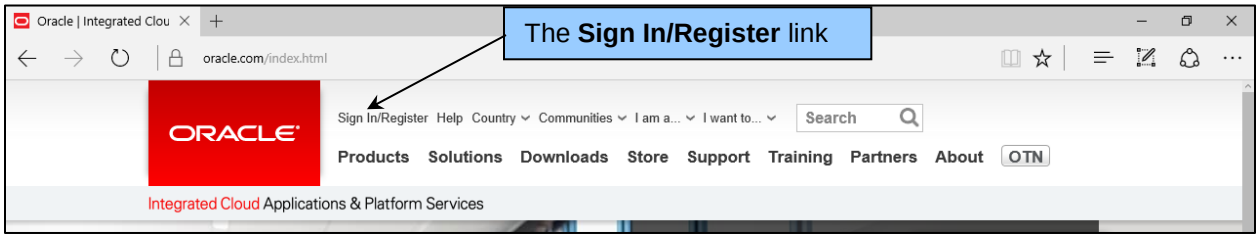


Figure B-1(a) — The Oracle Home Page

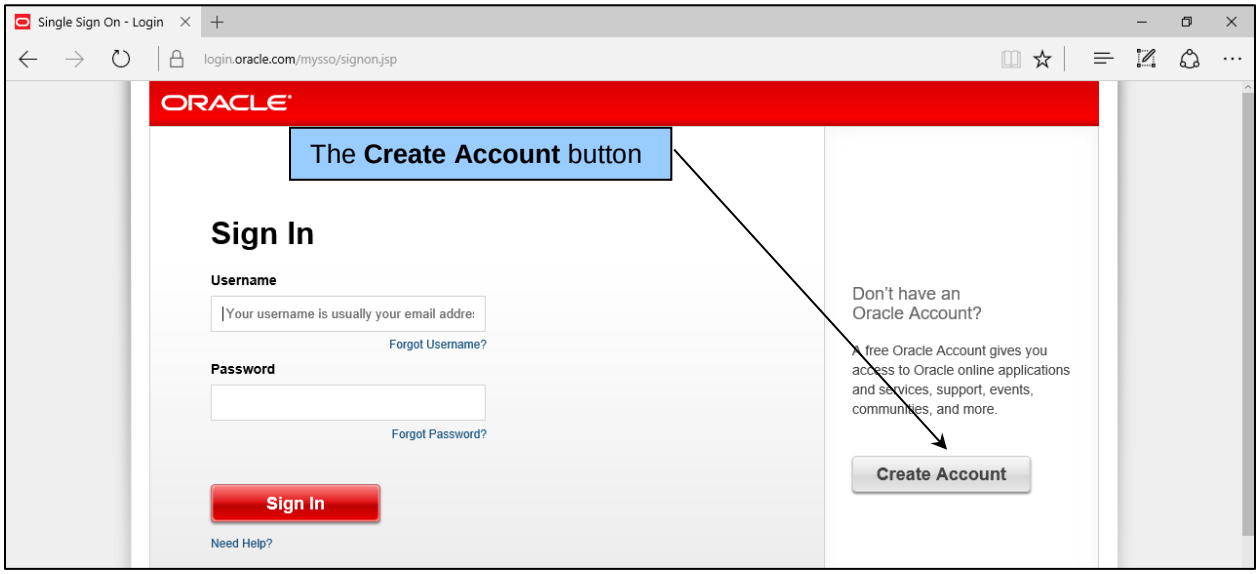


Figure B-1(b) — The Oracle Sign In/Register Page

Installing Oracle Database XE

We will step through the installation process for Oracle Database XE on a Windows operating system, specifically Windows 10, 64-bit edition.

1. Download the Oracle Database XE software: From the Oracle home page (www.oracle.com), hover the mouse over the **Downloads** link (see Figure B-2(a)) and from the **Database** section, click on **Oracle Database 11g Express Edition**. This will result in a window like Figure B-2(b). Click the "Accept License Agreement" button, then click on the proper file name to begin the download.
2. Extract the zipped file's contents to a folder—we created and used the folder **D:\OracleDownloadsDBC-e08-AppB**, but you can use any folder. The extraction will create a DISK1 folder (D:\OracleDownloadsDBC-e08-AppB\ODB_XE11R2\DISK1) within an ODB_XE11R2 folder and place all extracted folders and files in that folder.
3. To start the actual installation process, right-click the **setup.exe** file in the DISK1folder, and click **Run as administrator**.

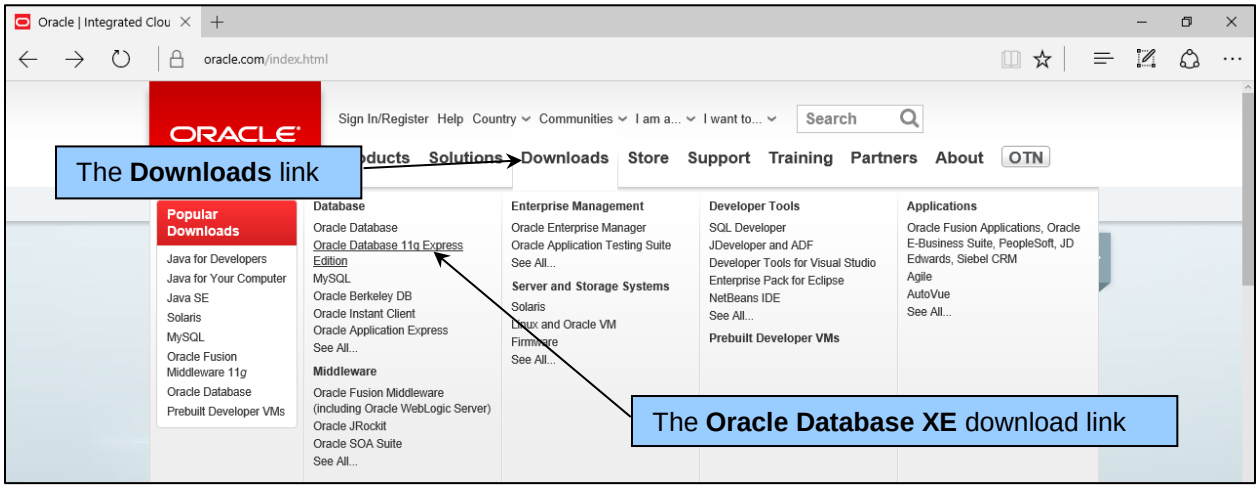


Figure B-2(a) — The Oracle Downloads Page

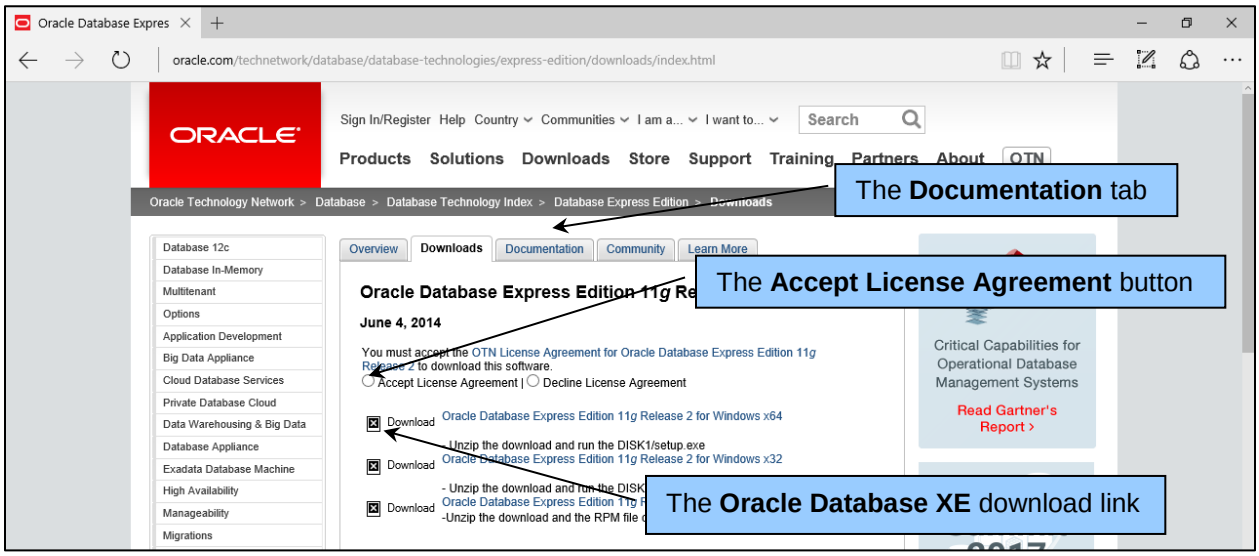


Figure B-2(b) — The Oracle Database XE Download Page

4. The **Oracle Database 11*g* Express Edition Install Wizard** is started and displays the **Welcome to the InstallShield Wizard for Oracle Database 11*g* Express Edition** screen, as shown in Figure B-3(a). **NOTE**: The exact name that Oracle software uses for Oracle Database XE is inconsistent and varies from place to place. We will write the name that is currently being used when it is displayed in utilities such as the Install Wizard but will consider *Oracle Database XE* as our preferred name in this appendix unless a variant is indicated.
5. Click the **Next** button. The **License Agreement** screen is displayed, as shown in Figure B-3(b).
6. Click the **I accept the terms in the license agreement** radio button to select it, and then click the **Next** button.

7. The **Choose Destination Location** screen is displayed, as shown in Figure B-3(c). The default location is C:\oraclexe, and we will use this location. If you want or need to install the program files in a different location, use the **Browse** button to select the alternate location.
8. Click the **Next** button. The **Specify Database Passwords** screen is displayed, as shown in Figure B-3(d). Enter the password that will be used for both the SYS and SYSTEM database user accounts.
9. Click the **Next** button. The **Summary** screen is displayed, as shown in Figure B-3(e). This screen lists installation settings and contains important data including file locations and TCP/IP port numbers. You should record this data for possible later use.
10. Click the **Install** button. The **Setup Status** screen is displayed during the installation and configuration of Oracle Database XE—this takes a while, so be patient. When the installation is complete, the **InstallShield Wizard Complete** screen is displayed, as shown in Figure B-3(f). If you examine your desktop, note that the installation process has added a desktop shortcut icon labeled *Get Started With Oracle Database 11g Express Edition*. This shortcut, also pictured in Figure B-3(f), is used to launch the **Oracle Database XE 11.2** Web administration utility, which we will discuss and use later in this appendix.
11. Click the **Finish** button to close the Oracle Database 11*g* Express Edition Install Wizard.

This completes the installation of Oracle Database XE. In order to make the best use of Oracle, however, we need an environment in which to develop applications (create tables, run queries, create users, etc.). The installation of Oracle Database XE includes **Oracle Application Express**, which is a Web-based development environment that can be used to build Oracle Database databases and applications. However, Oracle SQL Developer is more full-featured and more commonly used; in addition, it does not depend on a Web browser for its operation—it is software that needs to be installed, which is the topic of the next section.

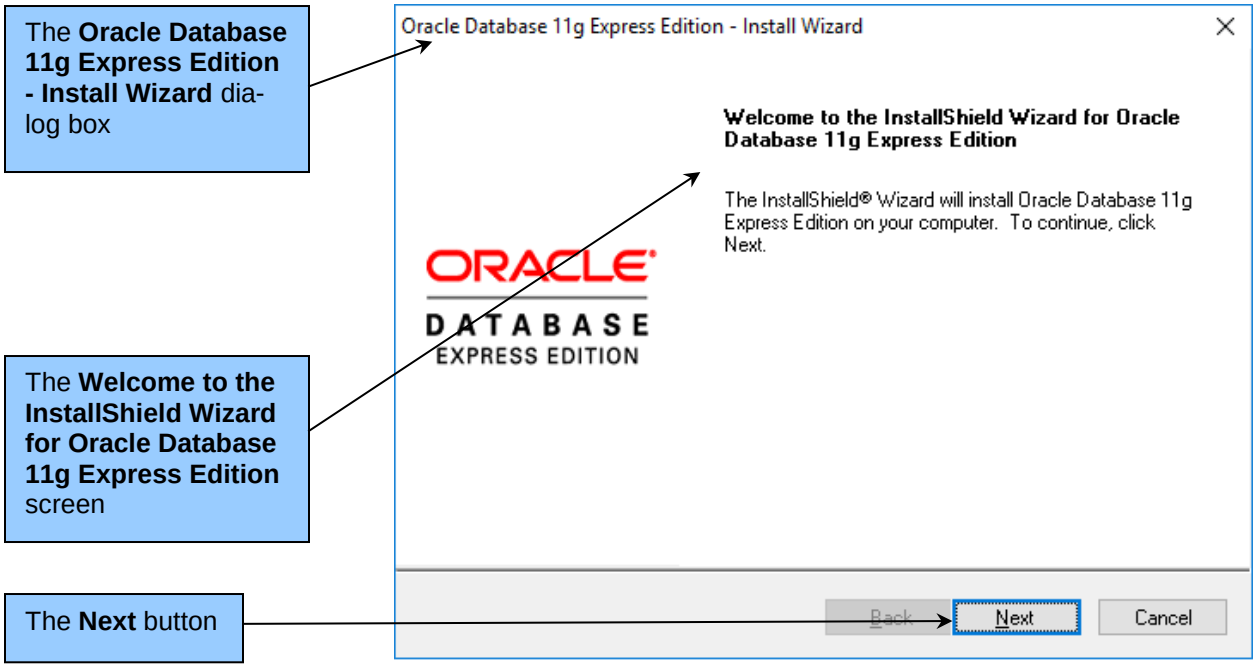


Figure B-3(a) — The Welcome to the InstallShield Wizard for Oracle Database 11g Express Edition Screen

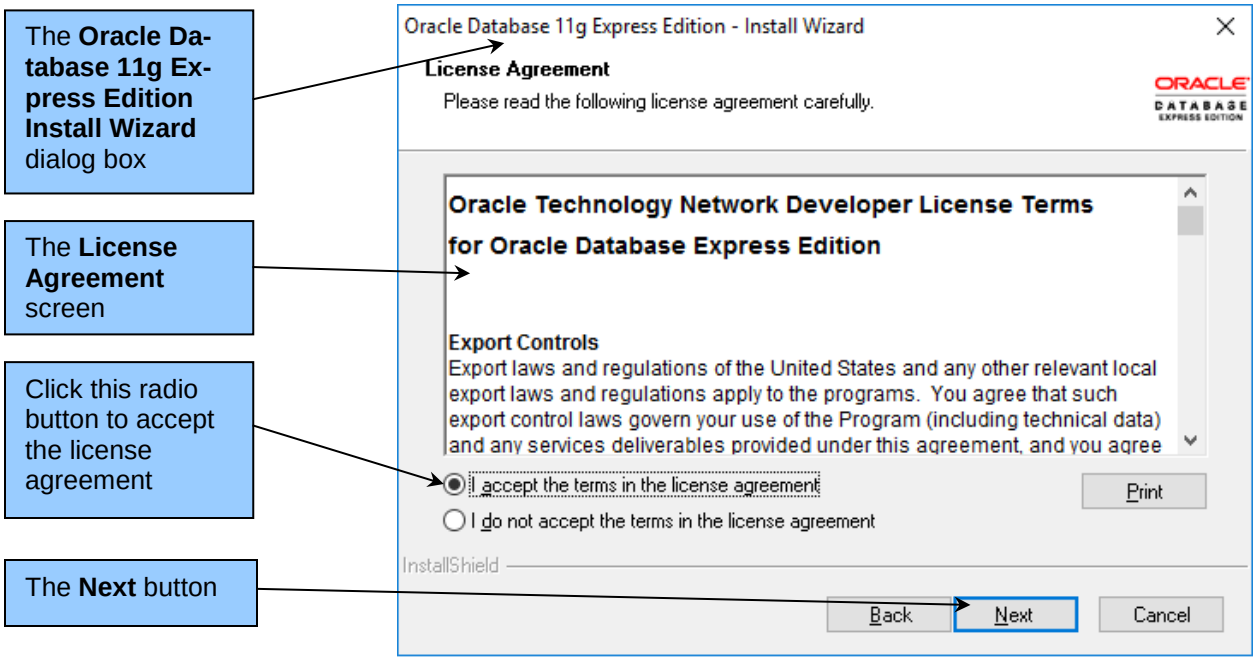


Figure B-3(b) — The License Agreement Screen

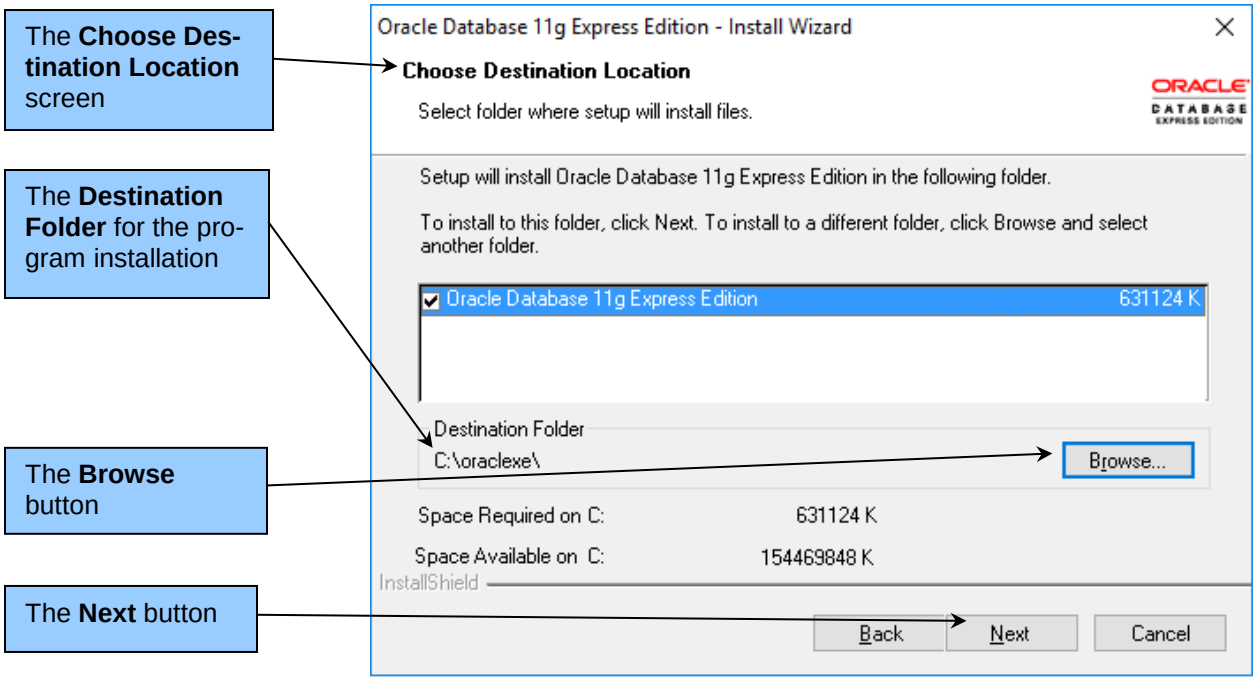


Figure B-3(c) — The Choose Destination Location Screen

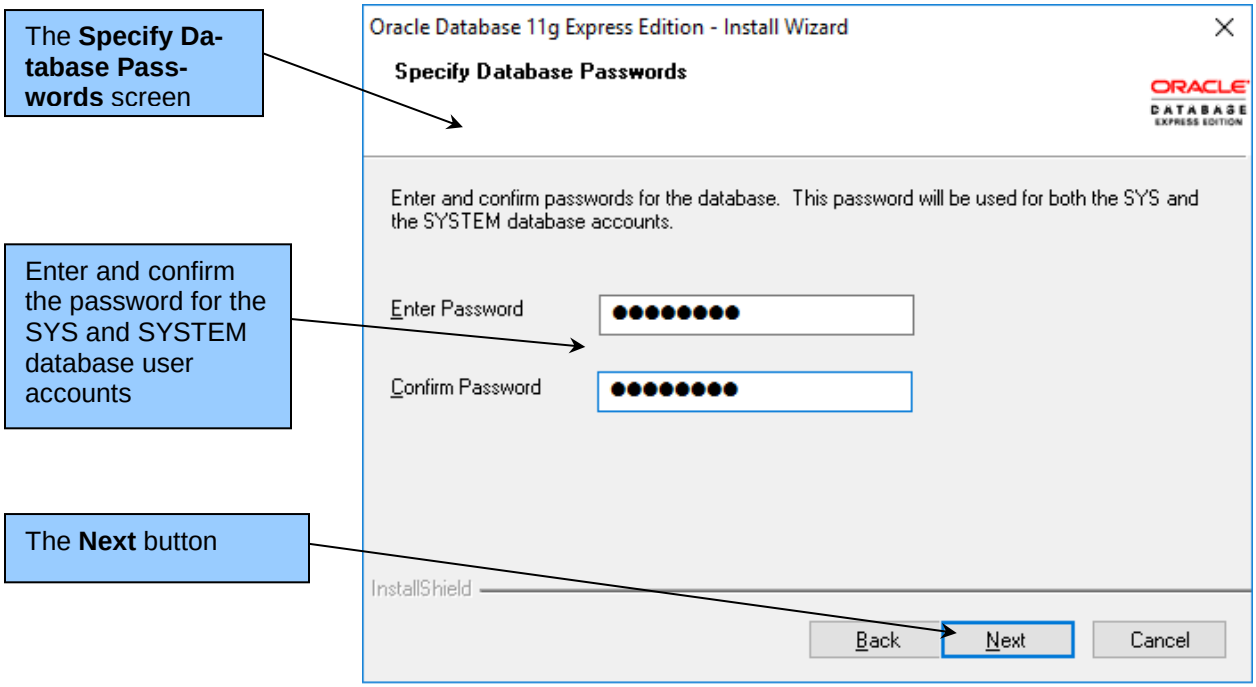


Figure B-3(d) — The Specify Database Passwords Screen

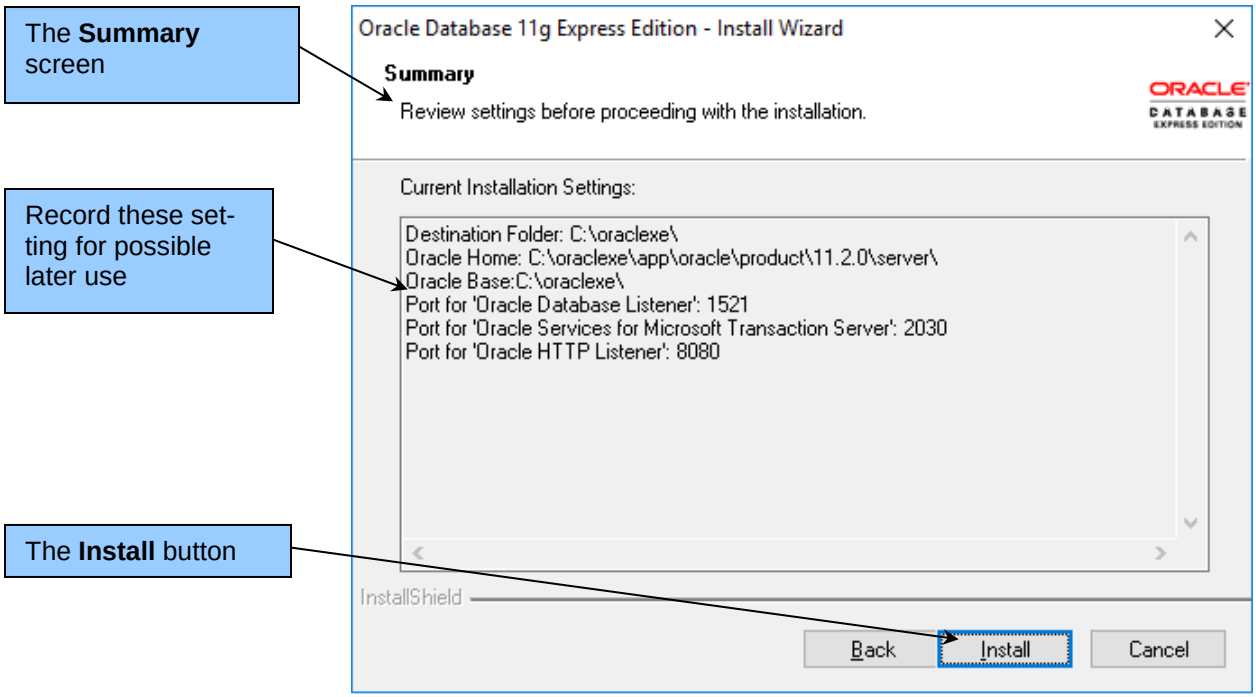


Figure B-3(e) — The Summary Screen

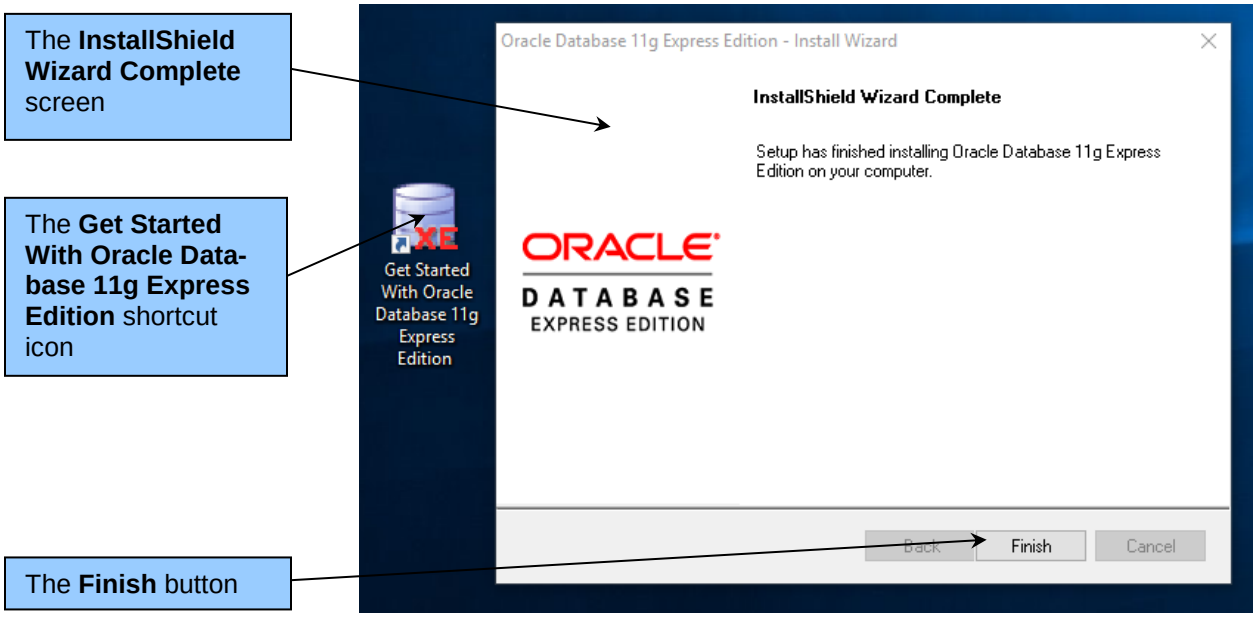


Figure B-3(f) — The InstallShield Wizard Complete Screen

Installing Oracle SQL Developer

Oracle SQL Developer is a GUI that allows creation of database objects (tables, functions, views, etc.) and provides an environment for writing and executing SQL queries. It has similar functionality to Microsoft SQL Server Management Studio (see Appendix A) and MySQL 5.7 Workbench (see Appendix C). Note that if you have installed Oracle Database 12*c*, SQL Developer was already installed as part of that, so you can skip this section.

1. Download the Oracle SQL Developer software: From the Oracle home page (www.oracle.com), hover the mouse over the **Downloads** link (see Figure B-4(a)) and from the **Developer Tools** section, click on **SQL Developer**. This will result in a window like Figure B-4(b). Click the "Accept License Agreement" button then click on the proper file name to begin the download. Since we are running a 64-bit version of Windows, we will select the **Windows 64-bit with JDK 8 included** file. If you are running a 32-bit Windows operating system, then you will need to install the most recent **Oracle Java Development Kit (JDK)** prior to continuing with the next step. To do this, follow the link to **JDK 8** shown in Figure B-4(b) to download and install JDK.

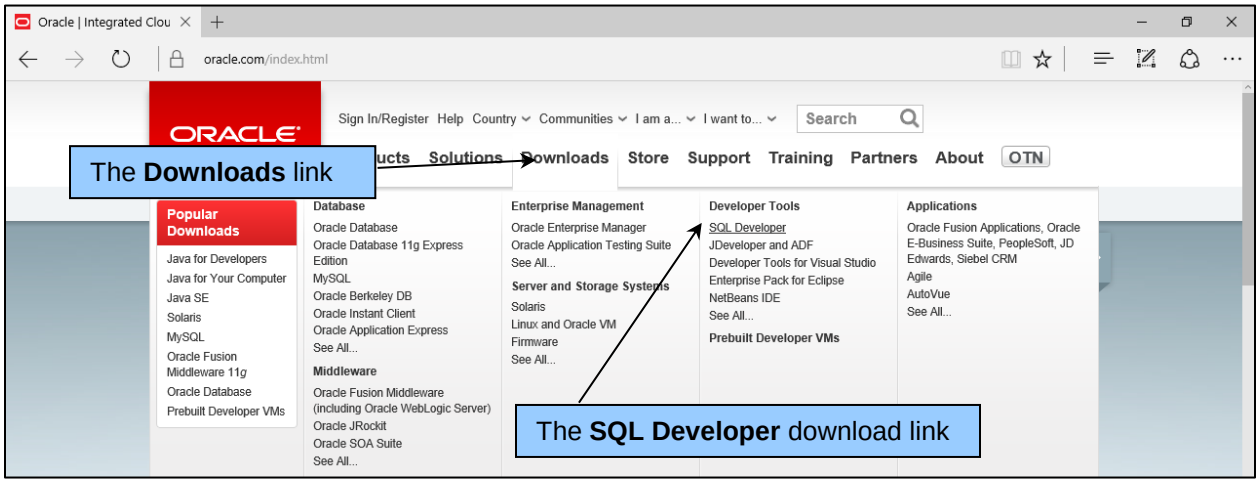


Figure B-4(a) — The Oracle Download Page

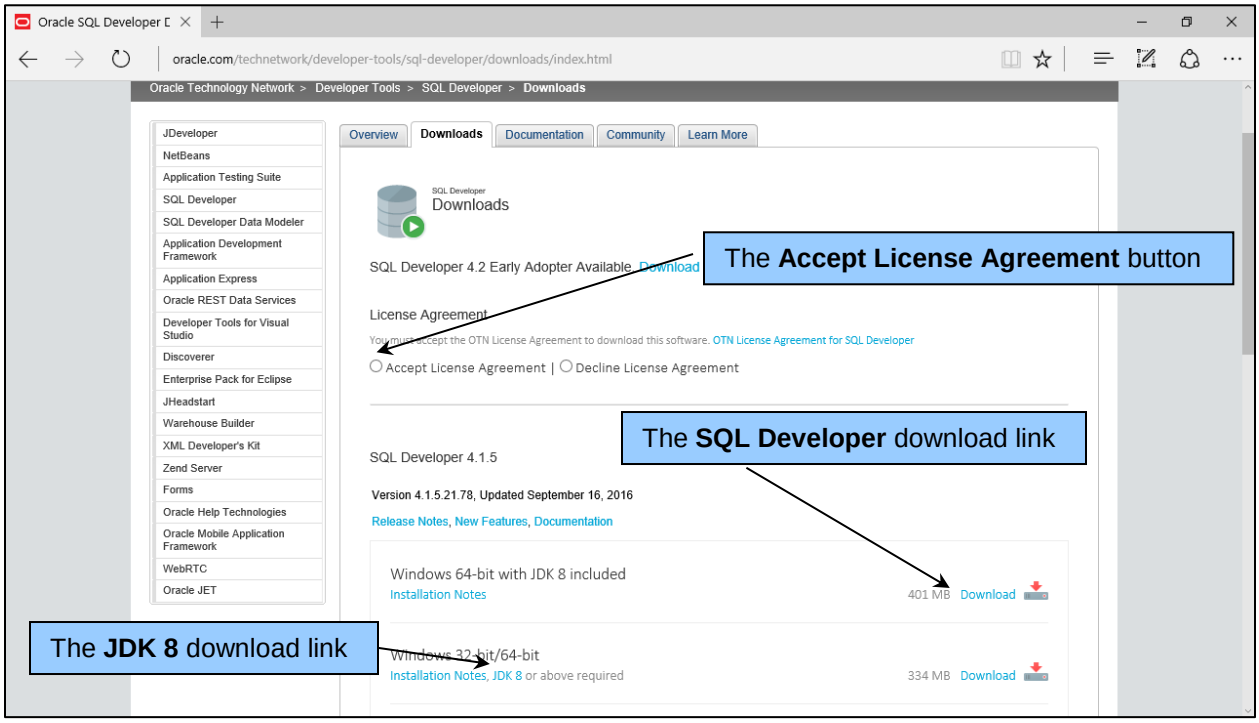


Figure B-4(b) — The Oracle SQL Developer Download Page

2. Extract the zipped file's contents to a folder—we created and used the folder **C:\SQLDevInstall**, but you can use any folder. The extraction will create a folder containing the SQL Developer software (in our case, C:\SQLDevInstall\sqldeveloper-4.1.5.21.78-x64) and place all extracted folders and files in that folder.
3. Before starting SQL Developer, it will be advisable to create a desktop shortcut for the executable file or to "pin" the icon to the task bar. Either approach will give you an easy way of launching the application. Within the folder created by the unzipping, there will be a folder called **sqldeveloper**. Go to this folder. To create a desktop shortcut for SQL Developer, right-click on the file "sqldeveloper.exe" and select **Create shortcut**, then move that shortcut to the desktop. To pin the application to the task bar instead, right-click and select **Pin to taskbar**.

We will use SQL Developer later in this appendix.

Oracle Database XE Administration and Development Tools

Oracle Database XE provides for both combined and separate administration and development tools. We prefer to do DBMS administration in the Web-based Oracle Database 11*g* XE utility and database development in the Oracle SQL Developer GUI utility.

Starting and Using the Oracle Database XE Utility

During the installation of Oracle Database XE, an icon labeled **Get Started with Oracle Database 11g Express Edition** was installed on the desktop, as shown in Figure B-3(f). Double-click this icon. At this point a dialog box that asks "The 'Java™ Plug-In SSV Helper' add-on from 'Oracle America, Inc.' is ready to use" may appear in some form depending on your Web browser, and if it does, click the **Enable** button.

We can now create the Oracle Database equivalent of a database. We will actually be creating a **workspace**, but it will function very similarly to a separate database in other DBMSs. To create the workspace, we have to create a Database Username, an Application Express Username, and a common password. We will create a user for the WP database described and used in Chapter 3.

1. Double-click the **Get Started with Oracle Database 11g Express Edition** icon on the desktop (as shown in Figure B-3(f)) to launch the Oracle Database XE Administration utility. The home page for the utility is shown in Figure B-5(a).
2. Click the **Application Express** button to bring up the login dialog box (Figure B-5(b)).
3. Enter the username SYSTEM and the password you created for the **SYSTEM account** during the installation process (you did write it down, right?). The SYSTEM account has DBA privileges, which we are about to use to create a database/user combination. Click the Login button to display the **Create Application Express Workspace** screen, as shown in Figure B-5(c).
4. Create the workspace by specifying **WP_USER** for both the *Database Username* and *Application Express Username*. Also specify **WP_USER+password** for the *Password*. Finally, click the **Create Workspace** button as shown in Figure B-5(c).
5. Figure B-5(d) shows the home page indicating successful creation of the WP_USER workspace. Click the **click here** link to get the login screen (not shown), which will have the *Workspace* and *Username* fields already filled in. Enter the password **WP_USER+password** (if it is not already filled in as well) then click the **Login** button on that screen.
6. The WP_USER workspace home page is displayed, as shown in Figure B-5(e). Although we will not use it in this book, Oracle Application Express is a Web-based application development environment that can be used for building Oracle Database databases and applications. Note, for example, the SQL Workshop tab, which provides an area for working with the SQL commands we discussed in Chapter 3. However, instead of using Oracle Application Express, we will use Oracle SQL Developer, discussed later in this appendix, as our database development tool.
7. Click the **Logout** link as shown in Figure B-5(e). We are done with the Application Express utility for now, so click the **X [Close]** button to close the Web browser.

The **Oracle Database XE 11.2** Web home page

Click the **Application Express** button

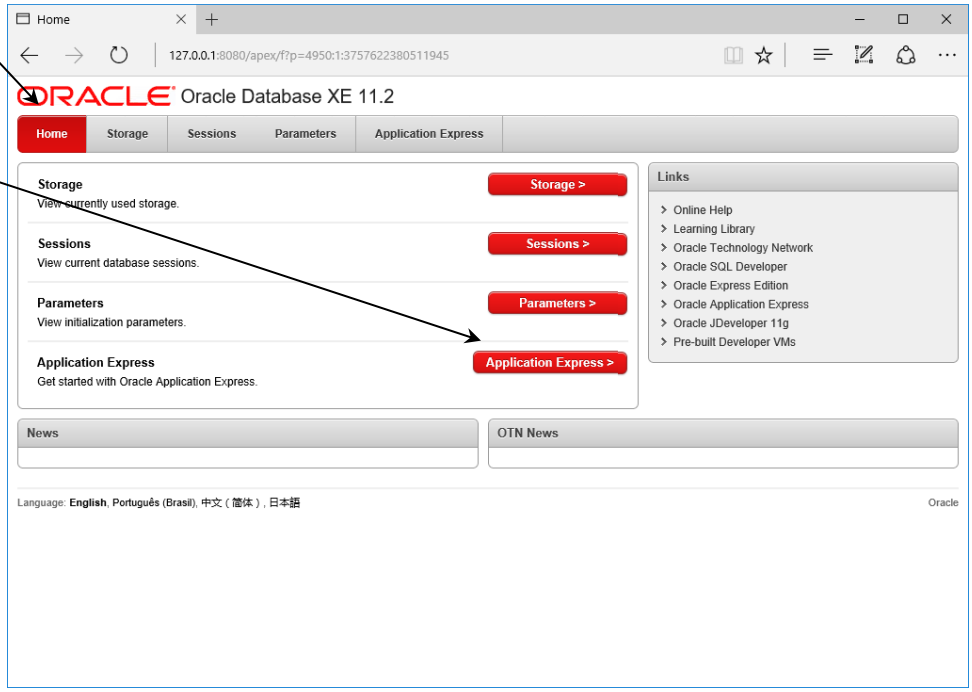


Figure B-5(a) — The Oracle Database XE 11.2 Web Home Page

The **Login** dialog box

Enter the **SYSTEM** Username and the password for the SYSTEM user that you created during installation

Click the **Login** button

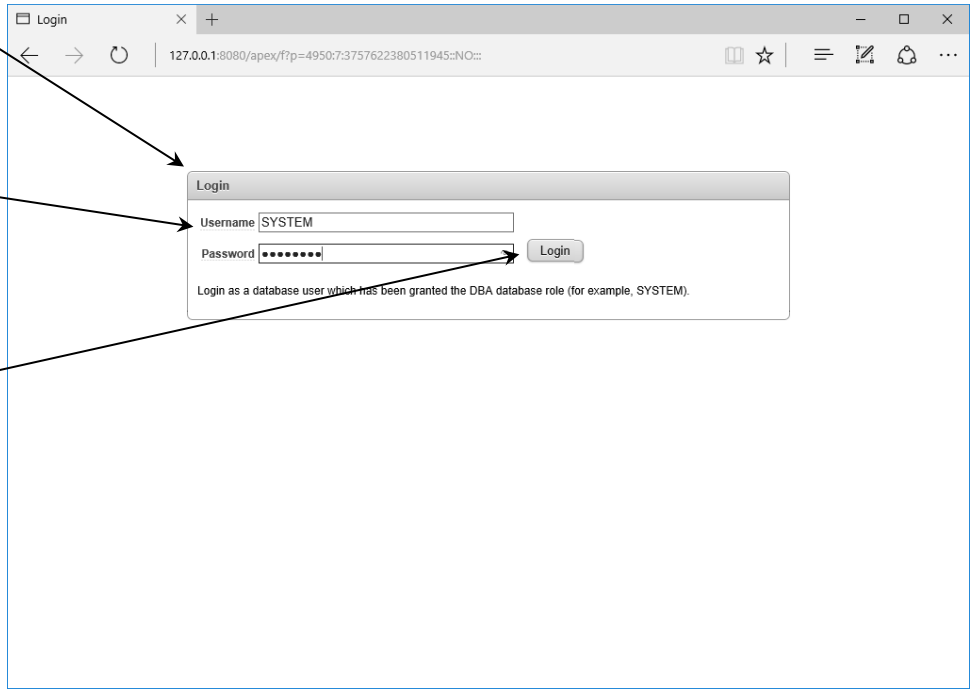


Figure B-5(b) — The Login Dialog Box

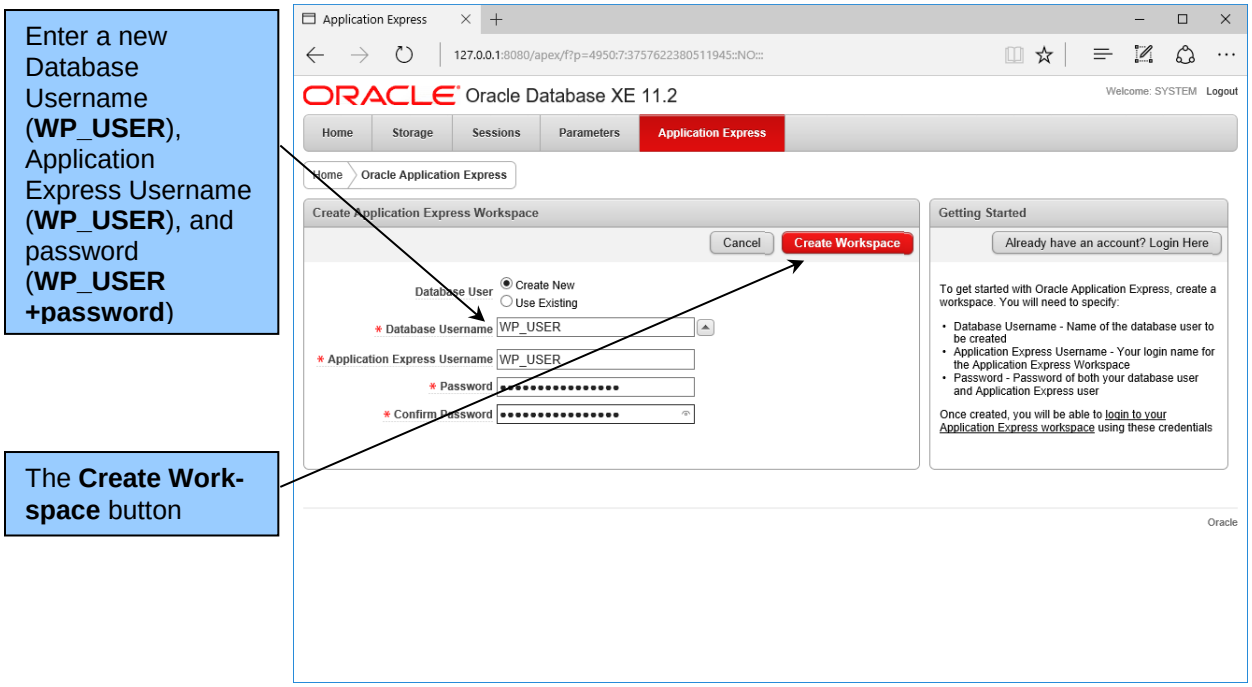


Figure B-5(c) — The Application Express Web Page

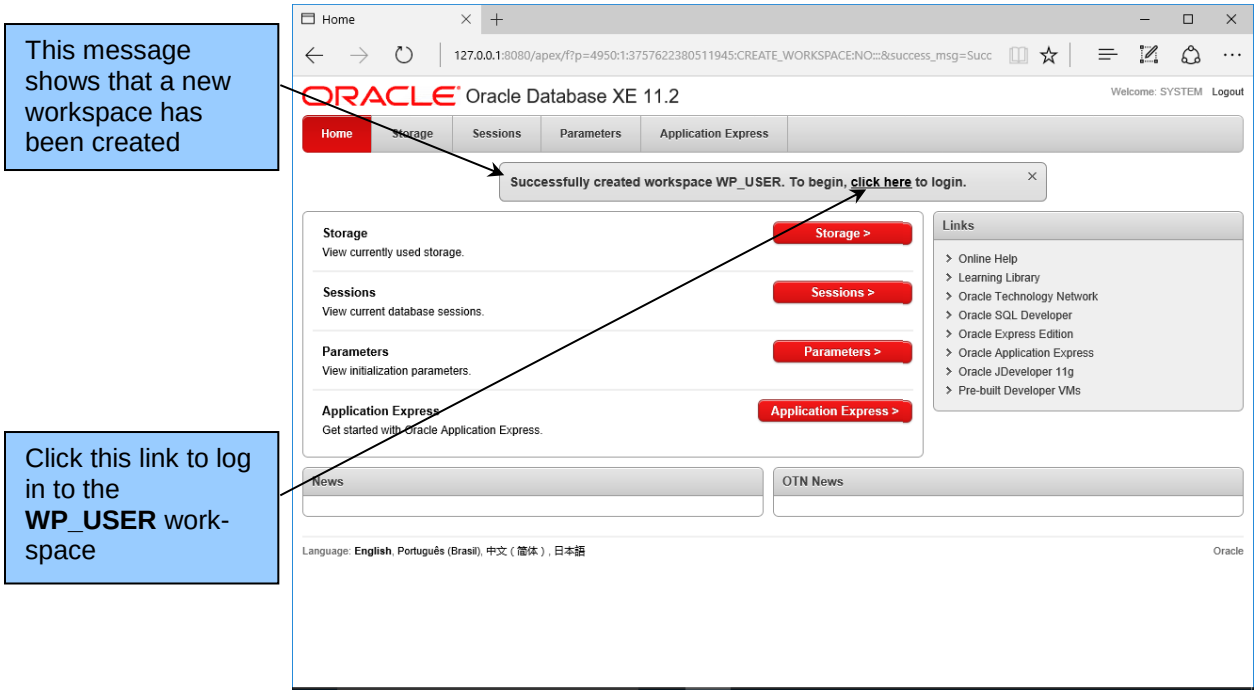


Figure B-5(d) — The Successfully Created Workspace Message

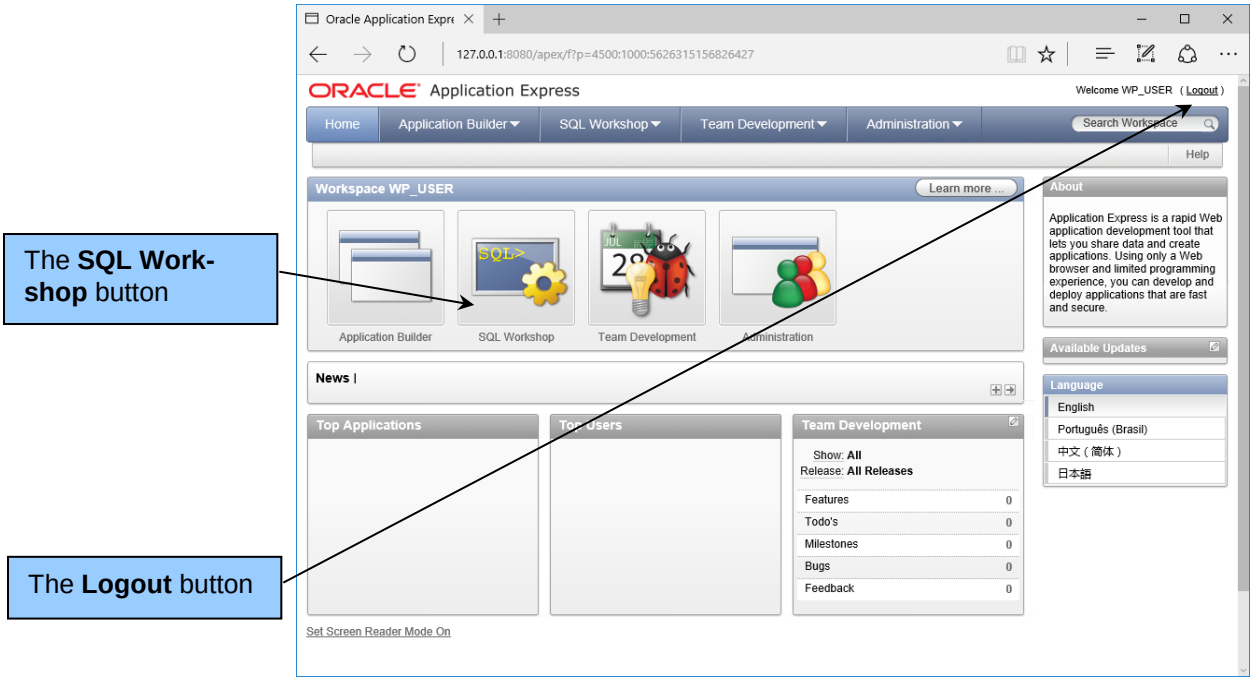


Figure B-5(e) — The WP_User Workspace Home Page

Starting and Using Oracle SQL Developer

Oracle has developed a very powerful GUI utility. The utility, Oracle SQL Developer, is particularly useful for database development with Oracle Database XE. SQL Developer is similar to the Microsoft SQL 2012 Server Management Studio (discussed in Appendix A) and the MySQL Workbench (discussed in Appendix C). The most recent version of SQL Developer can always be downloaded from the Oracle Web site (www.oracle.com) using the instructions found earlier in this appendix. In this appendix, we are running SQL Developer version 4.1.5.

To start Oracle SQL Developer, click the icon in the task bar (if you pinned it there after installation) or double-click the desktop shortcut icon (if you created one). If you did neither of those things, then you have to search for and double-click the *sqldeveloper.exe* program. When you start SQL Developer for the first time, a dialog box may prompt you (if you did not install the version with JDK included) for the location of the Java JDK *java.exe* file.[3] Be sure you have the JDK installed before opening SQL Developer! SQL Developer may also ask you to configure file type associations, and you should select all the appropriate types.

Our next task is to connect to the database using the WP_USER account in preparation for creating tables, inserting data, and running queries.

[3] On a Windows operating system, look in C:\Program Files\Java\jdk{version number}\bin.

1. If you have not already done so, start Oracle SQL Developer as described above.
2. Figure B-6(a) shows SQL Developer, where we have clicked the **New Connection** button to open the **New / Select Database Connection** dialog box.
3. We are using the settings needed to connect to our workspace using a **connection** named *WP*. In the top half of the connection window we specify who is connecting to the database. For the Connection Name, Username, and Password, enter the values you created earlier for the WP Workspace, as shown in Figure B-6(a).
4. In the bottom half of the connection window, we specify where SQL Developer should look to find the database. Note that when Oracle asks for a **System Identifier (SID)**, you can leave the default one that appears alone. In this scenario it is **xe.** You should also accept the default Host-name and Port values for now.
5. Click the **Connect** button. In Figure B-6(b), WP_USER has connected to the workspace using the WP connection, and we can see the WP database connection open in SQL Developer. Note that only one connection (WP) is present, but in general any number of connections may appear there. Also note that the list of object types in the red box is incomplete but does show the types of objects we are most interested in in this book.

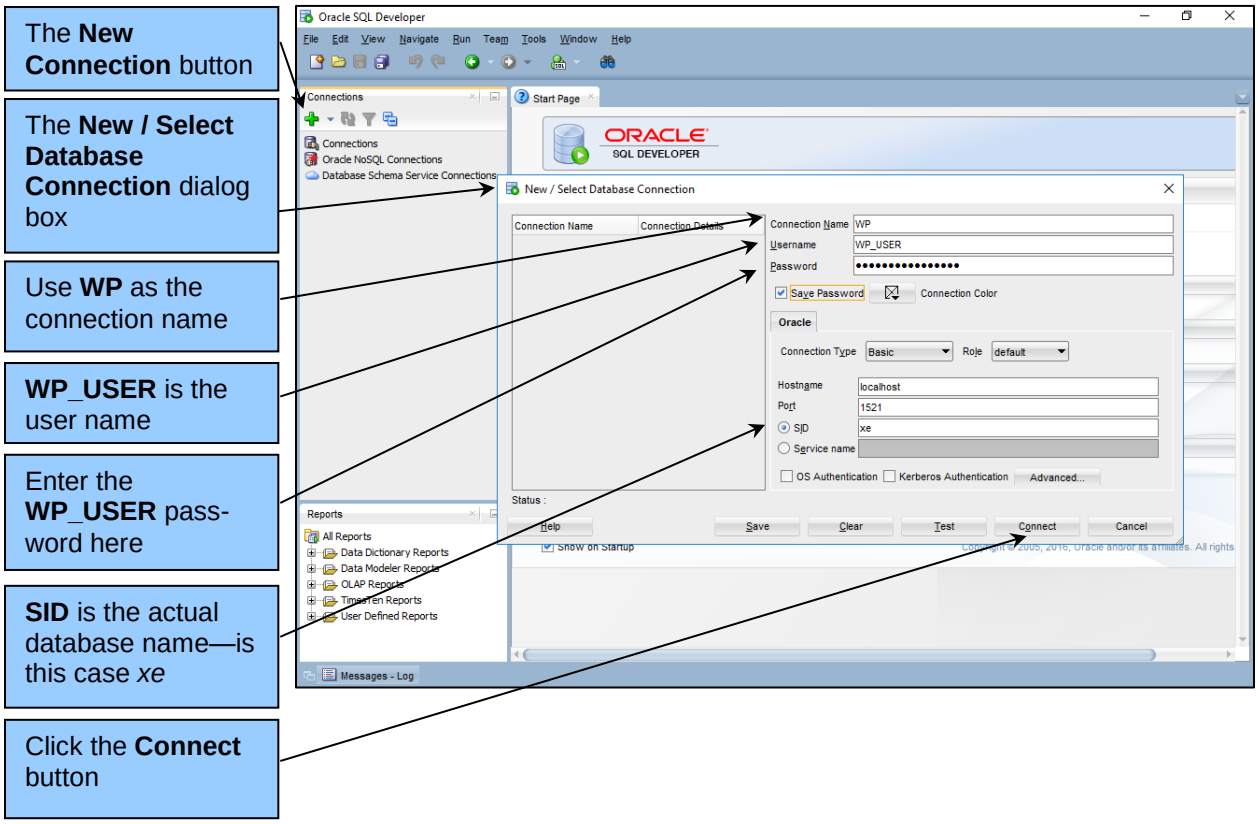


Figure B-6(a) — Connecting to the Database in Oracle SQL Developer

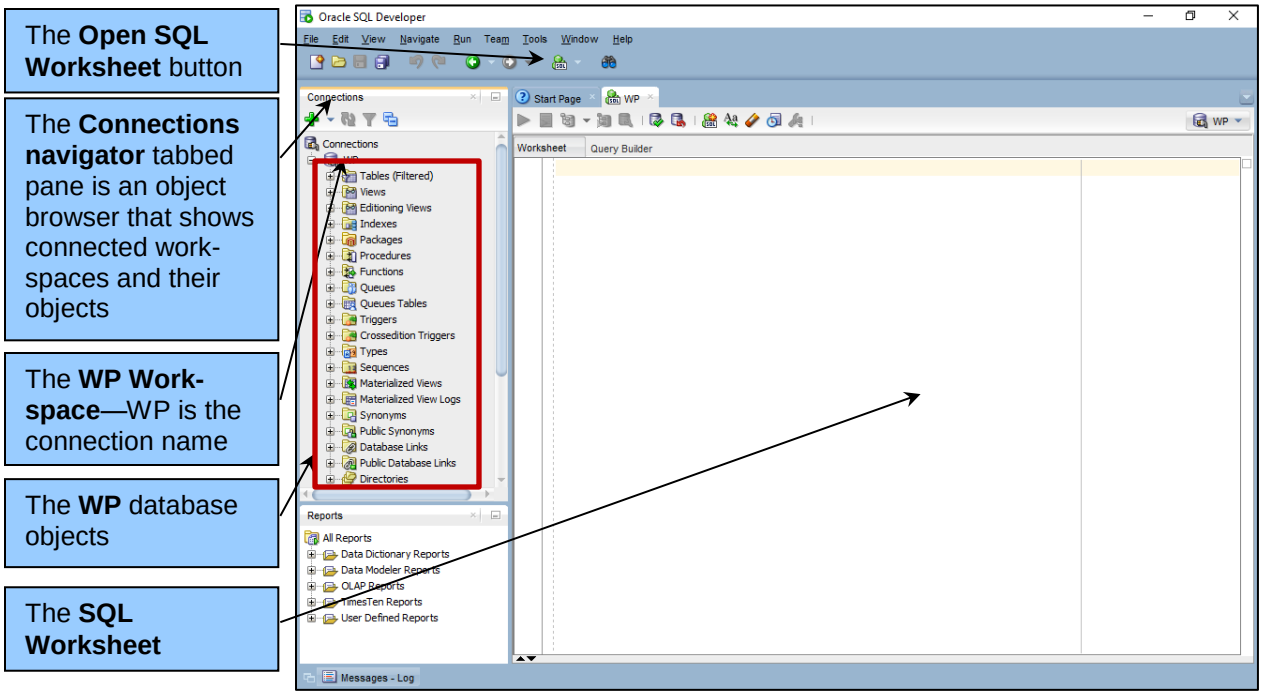


Figure B-6(b) — The WP Database in SQL Developer

Working with SQL Statements and Scripts in Oracle Database XE

In Oracle Database XE, 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. We will run our statements and scripts using Oracle SQL Developer (other environments, such as SQL*Plus, can also run statements and scripts). Later in this appendix we will use scripts to create tables and individual SQL statements to process single queries.

You can create scripts directly in Oracle SQL Developer or in any ASCII text editor. In the Windows operating system, the Notepad text editor is a good choice if you're not using Oracle SQL Developer itself. Regardless of which text editor you use, save your scripts with the file extension ***.sql** so that Oracle Database XE recognizes them. To organize your scripts, we recommend creating a new folder to hold the files for all your databases for this book and, within that folder, a separate new folder for each database. Figure B-7(a) shows the start of an SQL script and Figure B-7(b) shows the script being saved in a new folder for the WP database, which was created during the process of saving the file from within SQL Developer. (The folder containing this one was previously created in Windows.)

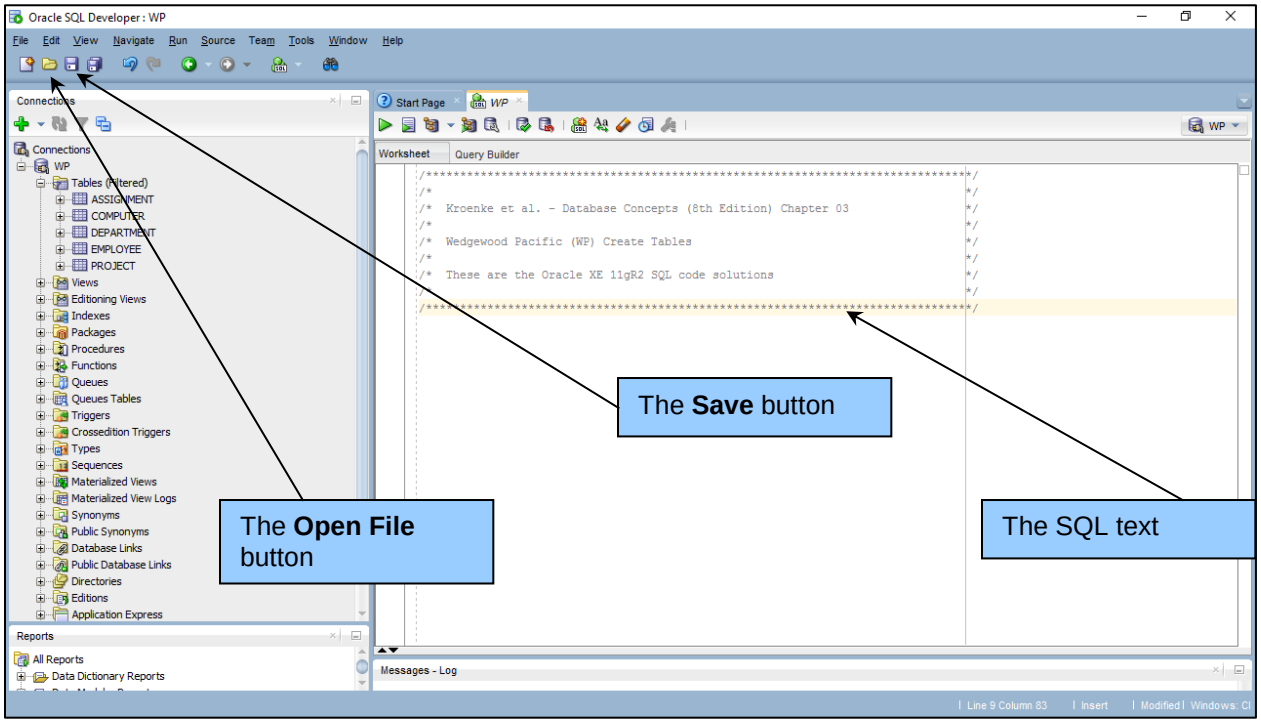


Figure B-7(a) — Entering SQL Statements in the SQL Editor

In Figure B-7(a), an SQL script can be opened from an existing operating system file using the **Open File** button or can be typed into the **SQL Worksheet** window. It can be saved using the **Save** button, which will bring up the window in Figure B-7(b), in which we first browsed to our main database folder

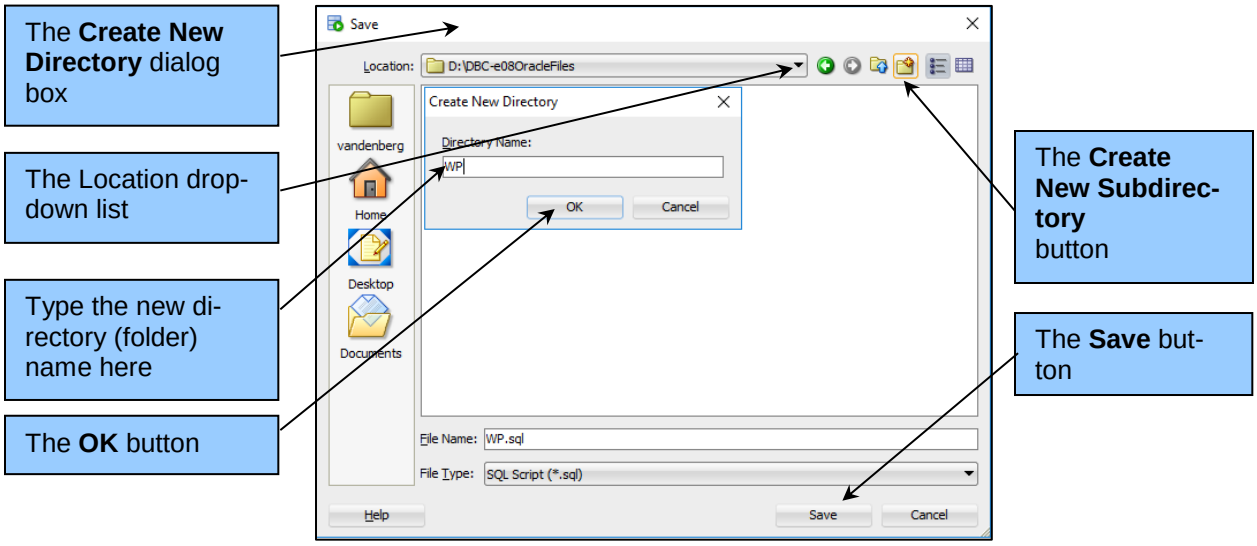


Figure B-7(b) — Creating the WP Database Directory (Folder)

(D:\DBC-e08OracleFiles) using the drop-down list, then used the Create New Subdirectory button to create the WP folder. Clicking **OK** then **Save** completes the process. Note that when you save or open a script, the name of the file appears on the SQL Worksheet tab to indicate you are working with a saved file.

How Do I Use SQL Commands to Create Table Structures?

The WP tabbed window that is already open in MySQL Developer is a script editor and, in fact, is the preferred script editor when we are working with Oracle Database XE databases. Figure B-8 shows the Oracle Database version of the SQL statements to create the WPC database shown in Figure 3-7 (and Oracle Database XE data types are shown in Chapter 3 in Figure 3-5(b)). Comparing the two versions of these SQL statements will help you understand any differences in how SQL is used in Oracle Database XE versus Microsoft SQL Server. The same statements are shown in SQL Developer in Figure B-9(a) together with some added header comments to document the script, saved as *DBC-e08-Oracle-WP-Create-Tables.sql*.

In Figure B-8, note the WP tables created by Oracle Database XE for this workspace and the WP sequences needed for use as primary keys. A **sequence** is an Oracle-supplied object that generates a sequential series of unique numbers. Sequences will be discussed further in the next section. For now, we focus on the steps needed to create and execute the script shown in Figure B-8:

1. In the tabbed SQL Worksheet document you saved above, which is now called WP.sql, type in the SQL statements shown in Figure B-8. You can also start a new SQL Worksheet for this purpose if you prefer. Maintain the spacing as shown in the figure to make it easier to read. The results should appear similar to Figure B-9(a).
2. We want to give this file a more descriptive name than "WP.sql." From the **File** menu, select the **Save As…** option and navigate to your WP folder, as described in the previous section.
3. Save the file as **DBC-e08-Oracle-WP-Create-Tables.sql** (use a shorter name if you prefer, but it should still be informative as to the purpose of the file).
4. Click the **Run Script button**, as shown in Figure B-9(a), to create the tables. If you are asked by Oracle SQL Developer to choose a connection in which to run these commands, select the **WP** connection. Alternatively, use the connection selector drop-down menu in the upper right to select the proper database connection (see Figure B-9(a)).
5. If there were any errors during the table creation, fix them, save the script again, then run it again, repeating this process until no errors remain.

Figure B-9(b) shows the newly created tables and sequences as they appear in the SQL Developer objects pane.

```
/*************************************************************************/
/*                                                                       */
/*     Kroenke et al. - Database Concepts (8th Edition) Chapter 03       */
/*                                                                       */
/*     Wedgewood Pacific [WP] Create Tables                              */
/*                                                                       */
/*  These are the Oracle Database XE 11g Release 2 SQL code solutions    */
/*                                                                       */
/*************************************************************************/

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

CREATE  TABLE EMPLOYEE(
    EmployeeNumber      Int              NOT NULL,
    FirstName           Char(25)         NOT NULL,
    LastName            Char(25)         NOT NULL,
    Department          Char(35)         DEFAULT 'Human Resources' NOT NULL,
    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)
    );
CREATE SEQUENCE seqEID Increment by 1 Start with 1;

CREATE  TABLE PROJECT(
    ProjectID           Int              NOT NULL,
    ProjectName         Char(50)         NOT NULL,
    Department          Char(35)         NOT NULL,
    MaxHours            Number(8,2)      DEFAULT 100 NOT NULL,
    StartDate           Date             NULL,
    EndDate             Date             NULL,
    CONSTRAINT          PROJECT_PK       PRIMARY KEY (ProjectID),
    CONSTRAINT          PROJ_DEPART_FK   FOREIGN KEY (Department)
                           REFERENCES DEPARTMENT(DepartmentName)
  );
CREATE SEQUENCE seqPID Increment by 100 Start with 1000;

CREATE  TABLE ASSIGNMENT (
    ProjectID           Int              NOT NULL,
    EmployeeNumber      Int              NOT NULL,
    HoursWorked         Number(6,2)      NULL,
    CONSTRAINT          ASSIGNMENT_PK    PRIMARY KEY(ProjectID, EmployeeNumber),
    CONSTRAINT          ASSIGN_PROJ_FK   FOREIGN KEY (ProjectID)
                           REFERENCES PROJECT (ProjectID)
                              ON DELETE CASCADE,
    CONSTRAINT          ASSIGN_EMP_FK    FOREIGN KEY (EmployeeNumber)
                           REFERENCES EMPLOYEE(EmployeeNumber)
  );
```

Figure B-8 — The WP Create Table SQL Statements for Oracle Database XE

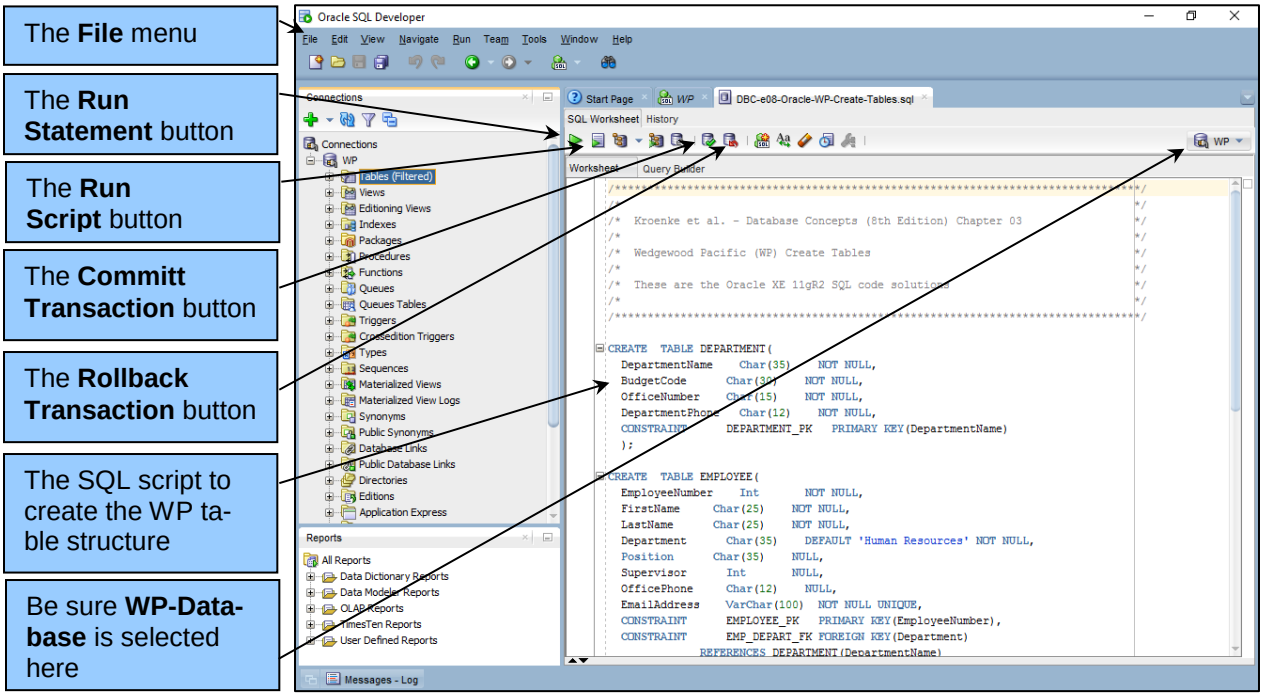


Figure B-9(a) — The SQL Script to Create the WP Table Structure

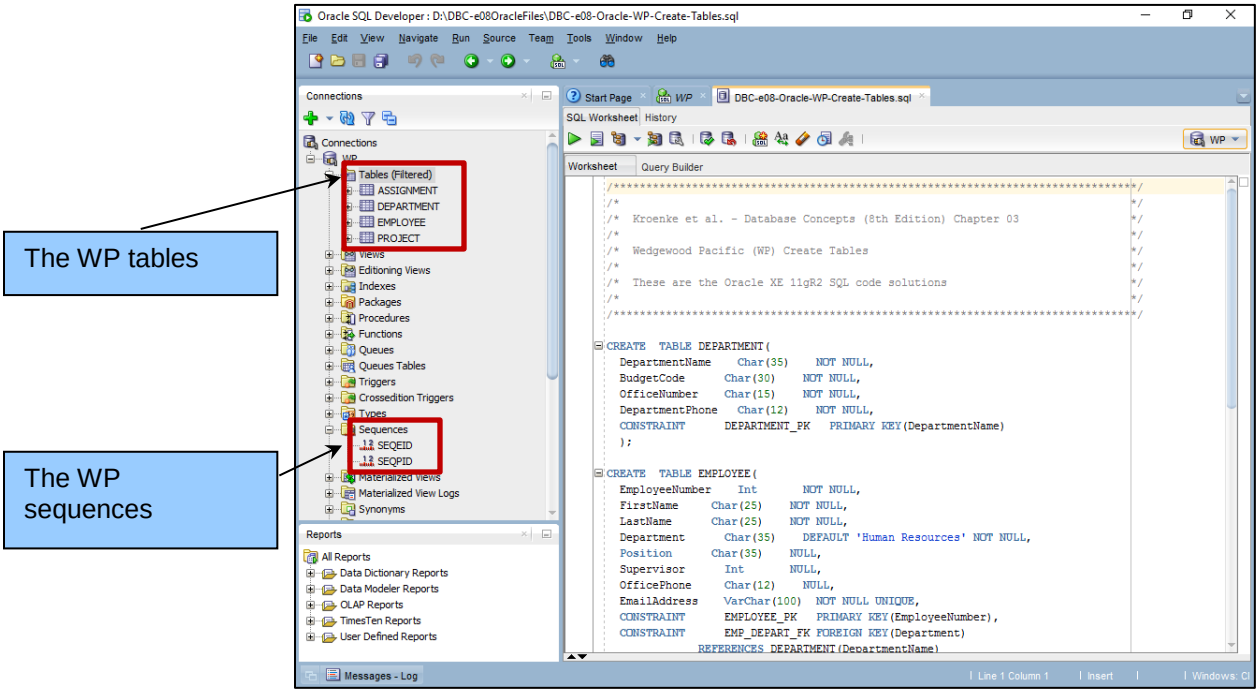


Figure B-9(b) — The WP Tables and Sequences in the Database

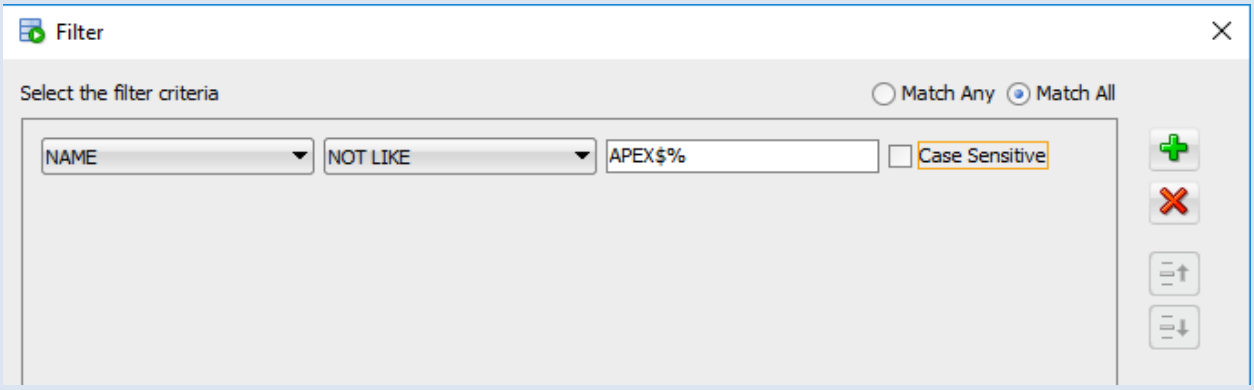
By The Way

Figure B-9(b) does **not** show the *metadata tables* Oracle Database XE creates for each workspace—these mostly have names beginning with "APEX$" and have been filtered out (see the next *By The Way* box below for how to do this). Also **not** shown are some *example tables* and *sequences* that are also created in each and every workspace. Unless you have a particular need for these tables and sequences, they should be deleted from the workspace to keep it clean and useable. Here is a script to delete the example tables and sequences—create an *.sql script using these SQL statements, save it, and then run it. You will have to refresh your view of the objects (right-click the *Tables* object to display the shortcut menu, then click *Refresh*—then do the same for the *Sequences* object) after you run the script.

```
/*****************************************************************/
/*                                                               */
/*  Kroenke et al. - Database Concepts (8th Edition) Appendix B  */
/*                                                               */
/*  SQL Script to drop unneeded Oracle XE tables and sequences   */
/*                                                               */
/*****************************************************************/

/* DROP unneeded example tables */

DROP TABLE DEMO_ORDER_ITEMS;
DROP TABLE DEMO_ORDERS;
DROP TABLE DEMO_PRODUCT_INFO;
DROP TABLE DEMO_STATES;
DROP TABLE DEMO_CUSTOMERS;
DROP TABLE DEMO_USERS;
DROP TABLE EMP;
DROP TABLE DEPT;

/* DROP unneeded example sequences */

DROP SEQUENCE DEMO_ORDER_ITEMS_SEQ;
DROP SEQUENCE DEMO_ORD_SEQ;
DROP SEQUENCE DEMO_PROD_SEQ;
DROP SEQUENCE DEMO_CUST_SEQ;
DROP SEQUENCE DEMO_USERS_SEQ;
```

By The Way

As mentioned above, Figure B-9(b) does **not** show the *metadata tables* Oracle Database XE creates for each workspace—these mostly are related to Oracle Application Express and have names beginning with "APEX$" and have been filtered out. To **filter** tables so that only the tables you want to see are listed, **right-click** the *Tables* object and select **Apply Filter …** To filter out the APEX$ tables, make your filter look like the image below, then click **OK**. Right-click the *Tables* object again and **refresh**, if necessary, to see just the tables you want to see.



How Do I Use SQL Statements to Insert Database Data?

You can enter data into Oracle Database XE either by entering data into a table grid in the SQL Developer GUI display or by using SQL INSERT statements. The SQL Developer GUI display is more useful for occasional data edits than for populating all the tables of a new database. You can open a table grid for data entry by opening the table in SQL Developer and then clicking the Data tab. There is an Insert Row button on this tab that enables you to add a new row if you need to do so. However, we will use the same method for populating the WP database tables that we used to create the table structure: an SQL script. Before we begin that task, however, we need to discuss two areas in which many relational DBMSs differ from each other: surrogate keys and dates.

Surrogate keys are handled in Oracle Database using Oracle objects called sequences. A sequence is an Oracle-supplied object that generates a sequential series of unique numbers. The following statement defines a sequence called seqEID that starts at 1 and is incremented by 1 each time it is used:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
Create Sequence seqEID Increment by 1 start with 1;
```

Two sequence methods are important to us. The method **NextVal** provides the next value in a sequence, and the method **CurrVal** provides the current value in a sequence. Thus, seqEID.NextVal provides the

next value of the seqEID sequence. You can insert a row into a table named EMPLOYEE using this sequence as follows:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
INSERT INTO EMPLOYEE VALUES (seqEID.NextVal, 'Mary', 'Jacobs',
   'Administration', 'CEO', NULL, '360-285-8110',
   'Mary.Jacobs@WP.com');
```

An EMPLOYEE row will be created with the next value in the sequence as the value for EmployeeID. Once this statement has been executed, you can retrieve the new row using CurrVal, as follows:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
SELECT      *
FROM        EMPLOYEE
WHERE       EmployeeID = seqEID.CurrVal;
```

Here, seqEID.CurrVal returns the current value of the sequence, which is the value just used.

Unfortunately, using sequences for surrogate keys has three problems. First, sequences can be used for purposes other than surrogate keys. Every time NextVal is called, a number is used up. If the value returned from NextVal is not used for an insert into a surrogate key column but is used for something else, then that value will be missing from the surrogate key range. A second, more serious, problem is that there is nothing in the schema that prevents someone from issuing an INSERT statement that does not use the sequence. Thus, Oracle Database accepts the following:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
INSERT INTO EMPLOYEE VALUES (123, 'Mary', 'Jacobs','Administration',
   'CEO', NULL, '360-285-8110', 'Mary.Jacobs@WP.com');
```

If this were done, duplicate values of a surrogate could occur. Third, it is possible that someone could accidentally use the wrong sequence when inserting into the table. If that were done, odd, erroneous, or duplicate surrogate key values would result. In spite of these possible problems, sequences are the recommended way for obtaining surrogate key values in Oracle.

One more point needs to be discussed. Entering values for Date data types can be problematic when using Oracle Database XE. Oracle wants dates in a particular format, but it is sometimes difficult to determine which format it wants. The TO_DATE function can be advantageous in such circumstances. **TO_DATE** takes two parameters, as shown here:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
TO_DATE('11/12/2016', 'MM/DD/YYYY')
```

The first parameter is the date value, and the second is the pattern to be used when interpreting the date. In this example, 11 is the month and 12 is the day of the month. You can use the TO_DATE function with the INSERT statement to provide date values for new rows. For example, suppose that table T1 has two columns—A and B—where A is an integer and B is a date; the following insert statement can be used:

```
/* *** EXAMPLE CODE – DO NOT RUN *** */
INSERT INTO T1 VALUES (100, TO_DATE ('01/05/16', 'DD/MM/YY');
```

The result will be a new row with the values 100 and the Oracle internal format for May 1, 2016. TO_DATE can also be used with UPDATE statements.

Figure B-10 shows the SQL statements to insert the data into the WP database. Note the COMMIT statements—Oracle Database requires these to finalize the data in the tables. Alternatively, you can click on the **Commit** button in SQL Developer to accomplish the same thing, as shown in Figure B-9(a). This is also a good time to mention the **Rollback** button (see Figure B-9(a) again), which will **rollback** (undo) all data changes (insert, update, or delete) performed since the most recent **commit** (be aware that Oracle also sometimes commits without being asked to, such as when any table is created).

Now that you understand some of those nuances, you are ready to insert the data. As with the CREATE TABLE statements in the previous section, type these insert statements into an SQL Worksheet window in Oracle SQL Developer (or use another application then open the file in Oracle SQL Developer). Save the file using a descriptive file name (we chose **DBC-e08-Oracle-WP-Insert-Data.sql**). Be sure that the WP database is still the active connection (see Figure B-9(a)), then run the script using the **Run Script** button. As with the table creation process, if there are any errors, fix them, save, and run again until everything is inserted properly.

Figure B-10 — The WP Data SQL Statements for Oracle Database XE

```
/*****   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-800-10', 'BLDG02-250', '360-287-8700');
INSERT INTO DEPARTMENT VALUES(
       'Production', 'BC-900-10', 'BLDG02-110', '360-287-8800');

COMMIT;

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

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

Figure B-10 Continued — The WP Data SQL Statements for Oracle Database XE

```
INSERT INTO EMPLOYEE VALUES(seqEID.NextVal,
       'George', 'Jones', 'Production', 'OPS2', 15,
       '360-287-8830', 'George.Jones@WP.com');
INSERT INTO EMPLOYEE(EmployeeNumber, FirstName, LastName, Department, Position,
       Supervisor, EmailAddress)
        VALUES(seqEID.NextVal,
       'Julia', 'Hayakawa', 'Production', 'OPS1', 15, 'Julia.Hayakawa@WP.com');
INSERT INTO EMPLOYEE(EmployeeNumber, FirstName, LastName, Department, Position,
       Supervisor, EmailAddress)
        VALUES(seqEID.NextVal,
       'Sam', 'Stewart', 'Production', 'OPS1', 15, 'Sam.Stewart@WP.com');

COMMIT;

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

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

COMMIT;

/*****   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);
```

Figure B-10 Continued — The WP Data SQL Statements for Oracle Database XE

```
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);
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);

COMMIT;

/******************************************************************************/
```

Figure B-10 Continued — The WP Data SQL Statements for Oracle Database XE

How Do I Work with SQL Queries in Oracle Database XE?

Now that we've created and populated the 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.

By now you should be familiar with using the SQL query tabbed document window and running SQL scripts. Individual SQL commands are typed in the same SQL Worksheet windows as you used for the scripts. Another way to open a worksheet for a database is to right-click the connection name and select the **Open SQL Worksheet** option. So, to create and run an SQL query:

1. Open an SQL Worksheet for the WP database using any of the methods previously mentioned.
2. Type the text of the SQL query you want to run (see Figure B-11 for a suggestion).

3. Be sure the WP database is selected (see the callout in Figure B-11)
4. Make sure the cursor is on a line containing part of the query, then click the **Run Statement button** in the SQL Editor toolbar (see Figure B-11).
5. 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 B-11. You can adjust the sizes 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 SQL Worksheet button again opens another tabbed query window. For more information about running queries in Oracle Database XE, see the Chapter 3 discussion beginning on page 159. Figure B-11 is similar to Figure 3-14 and shows an SQL query on the WP database being run in Oracle SQL Developer.

It is also possible to have multiple SQL queries in a single SQL Worksheet window. The **Run Script** button will run all the queries, with output from all the queries appearing sequentially in a **Script Output** tab (rather than a more graphical **Query Result** tab). To run just a single query in a worksheet containing multiple queries, simply position the cursor anywhere in that query and click the **Run Statement** button and the results will appear in a **Query Result** tab.

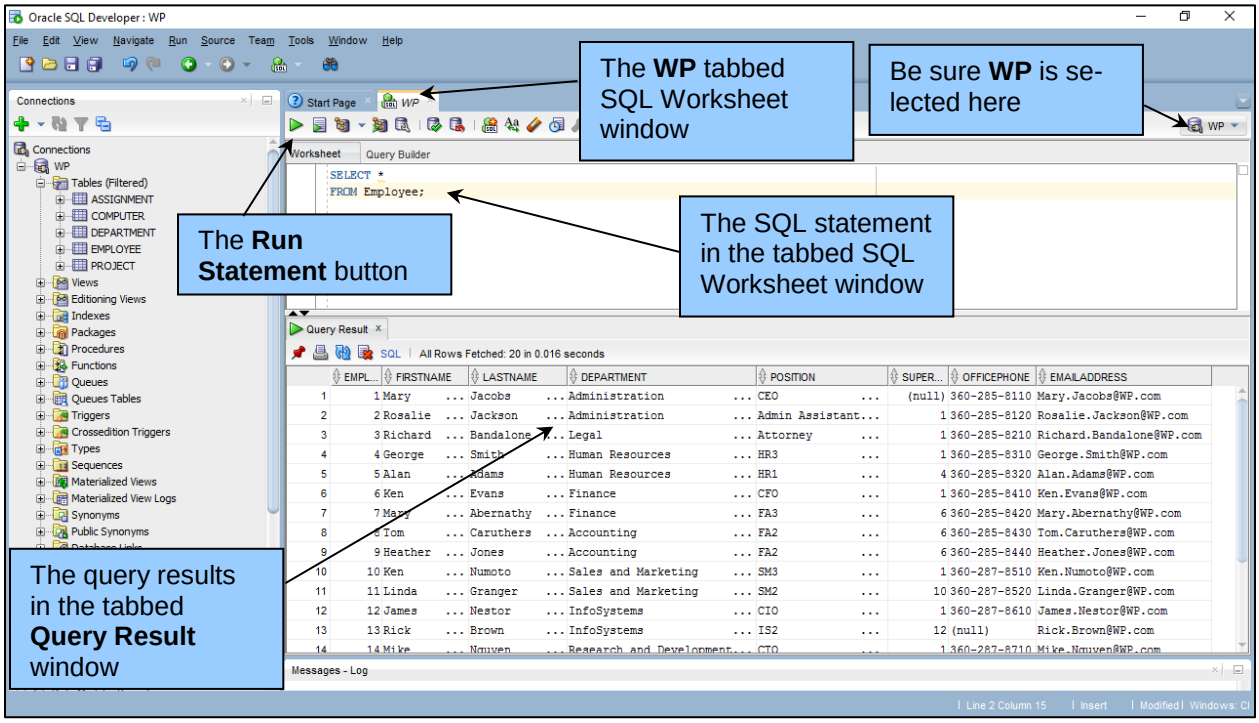


Figure B-11 — Running an SQL Query in Oracle SQL Developer

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 an Oracle Database XE 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 and 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 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)**.

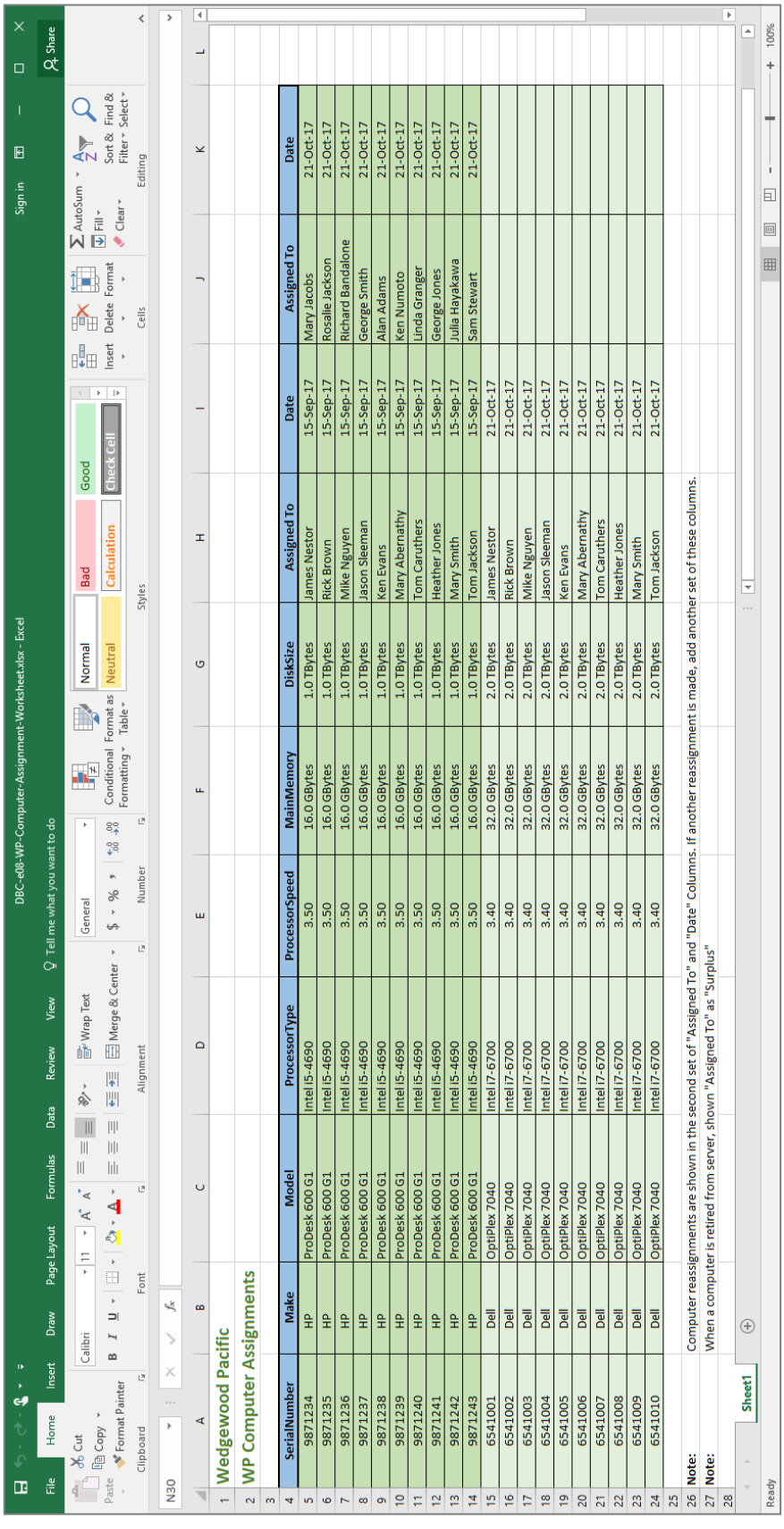
As an example, let's consider the problem of computers owned by WP. WP needs to track these computers (asset inventory) and who they are currently and have been assigned to for use. The properly designed tables (COMPUTER and COMPUTER_ASSIGNMENT) to handle this problem are shown in the Chapter 3 Access 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 B-12. 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:

$$\textbf{EmployeeID} \rightarrow\!\!\!\!\rightarrow \textbf{PhoneNumber}$$



Wedgewood Pacific

WP Computer Assignments

SerialNumber	Make	Model	ProcessorType	ProcessorSpeed	MainMemory	DiskSize	Assigned To	Date	Assigned To	Date
9871234	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	James Nestor	15-Sep-17	Mary Jacobs	21-Oct-17
9871235	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Rick Brown	15-Sep-17	Rosalie Jackson	21-Oct-17
9871236	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Mike Nguyen	15-Sep-17	Richard Bandalone	21-Oct-17
9871237	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Jason Sleeman	15-Sep-17	George Smith	21-Oct-17
9871238	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Ken Evans	15-Sep-17	Alan Adams	21-Oct-17
9871239	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Mary Abernathy	15-Sep-17	Ken Numoto	21-Oct-17
9871240	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Tom Caruthers	15-Sep-17	Linda Granger	21-Oct-17
9871241	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Heather Jones	15-Sep-17	George Jones	21-Oct-17
9871242	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Mary Smith	15-Sep-17	Julia Hayakawa	21-Oct-17
9871243	HP	ProDesk 600 G1	Intel I5-4690	3.50	16.0 GBytes	1.0 TBytes	Tom Jackson	15-Sep-17	Sam Stewart	21-Oct-17
6541001	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	James Nestor	21-Oct-17		
6541002	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Rick Brown	21-Oct-17		
6541003	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Mike Nguyen	21-Oct-17		
6541004	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Jason Sleeman	21-Oct-17		
6541005	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Ken Evans	21-Oct-17		
6541006	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Mary Abernathy	21-Oct-17		
6541007	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Tom Caruthers	21-Oct-17		
6541008	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Heather Jones	21-Oct-17		
6541009	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Mary Smith	21-Oct-17		
6541010	Dell	OptiPlex 7040	Intel I7-6700	3.40	32.0 GBytes	2.0 TBytes	Tom Jackson	21-Oct-17		

Note: Computer reassignments are shown in the second set of "Assigned To" and "Date" Columns. If another reassignment is made, add another set of these columns.

Note: When a computer is retired from server, shown "Assigned To" as "Surplus"

Figure B-12 – The WP Computer Assignments in a Microsoft Excel 2016 Worksheet

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
- Modifying the structure and data in each worksheet so that it is correct for importing into the database, then
- Importing 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 keys, foreign keys, and any other needed constraints—while we will use the SQL ALTER TABLE statement as needed here, for a more complete discussion of the SQL ALTER TABLE statement, see Appendix E, "Advanced SQL".

Preparing the Microsoft Excel Data for Import into a Database Table

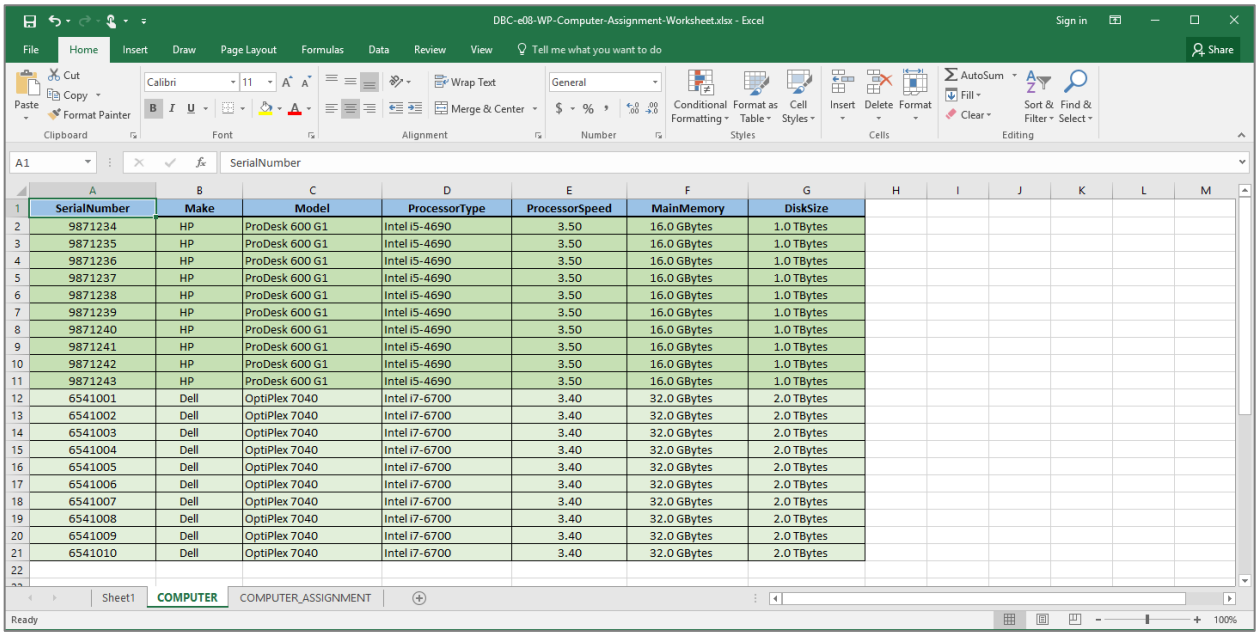
Figure B-13 shows the COMPUTER worksheet after it has been cleaned up. All extraneous rows and columns have been deleted, and only the computer data (with appropriate column headers) remains. This worksheet now looks like a database table, which is a good indication that the data import should work properly.

Figure B-14 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 WPC database:

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

This query gives us:

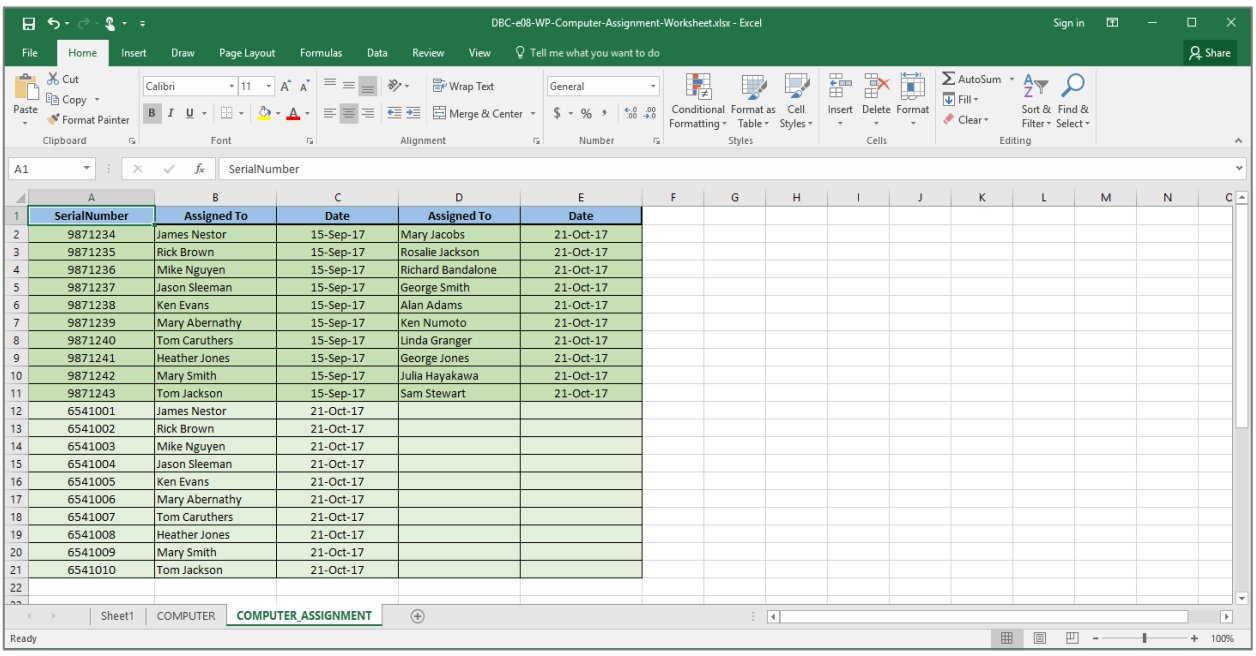
[4] For more information about multivalued dependencies and the multivalue, multicolumn problem, see David Kroenke and David Auer, *Database Processing: Fundamentals, Design, and Implementation*, 14th ed. (Upper Saddle River, NJ: Pearson Higher Education, 2016).



DBC-e08-WP-Computer-Assignment-Worksheet.xlsx - Excel

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

Figure B-13 — The WP COMPUTER Worksheet



DBC-e08-WP-Computer-Assignment-Worksheet.xlsx - Excel

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 B-14 — The WP COMPUTER_ASSIGNMENT Worksheet – First Attempt

	EMPL...	FIRSTNAME	LASTNAME	DEPARTMENT	POSITION	SUPER...	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	Abernathy	Finance	FA3	6	360-285-8420	Mary.Abernathy@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	360-287-8510	Ken.Numoto@WP.com
11	11	Linda	Granger	Sales and Marketing	SM2	10	360-287-8520	Linda.Granger@WP.com
12	12	James	Nestor	InfoSystems	CIO	1	360-287-8610	James.Nestor@WP.com
13	13	Rick	Brown	InfoSystems	IS2	12	(null)	Rick.Brown@WP.com
14	14	Mike	Nguyen	Research and Development	CTO	1	360-287-8710	Mike.Nguyen@WP.com
15	15	Jason	Sleeman	Research and Development	RD3	14	360-287-8720	Jason.Sleeman@WP.com
16	16	Mary	Smith	Production	OPS3	1	360-287-8810	Mary.Smith@WP.com
17	17	Tom	Jackson	Production	OPS2	14	360-287-8820	Tom.Jackson@WP.com
18	18	George	Jones	Production	OPS2	15	360-287-8830	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 B-15, which also includes a renamed *Date* column which is now *DateAssigned.* Admittedly this is a small example,

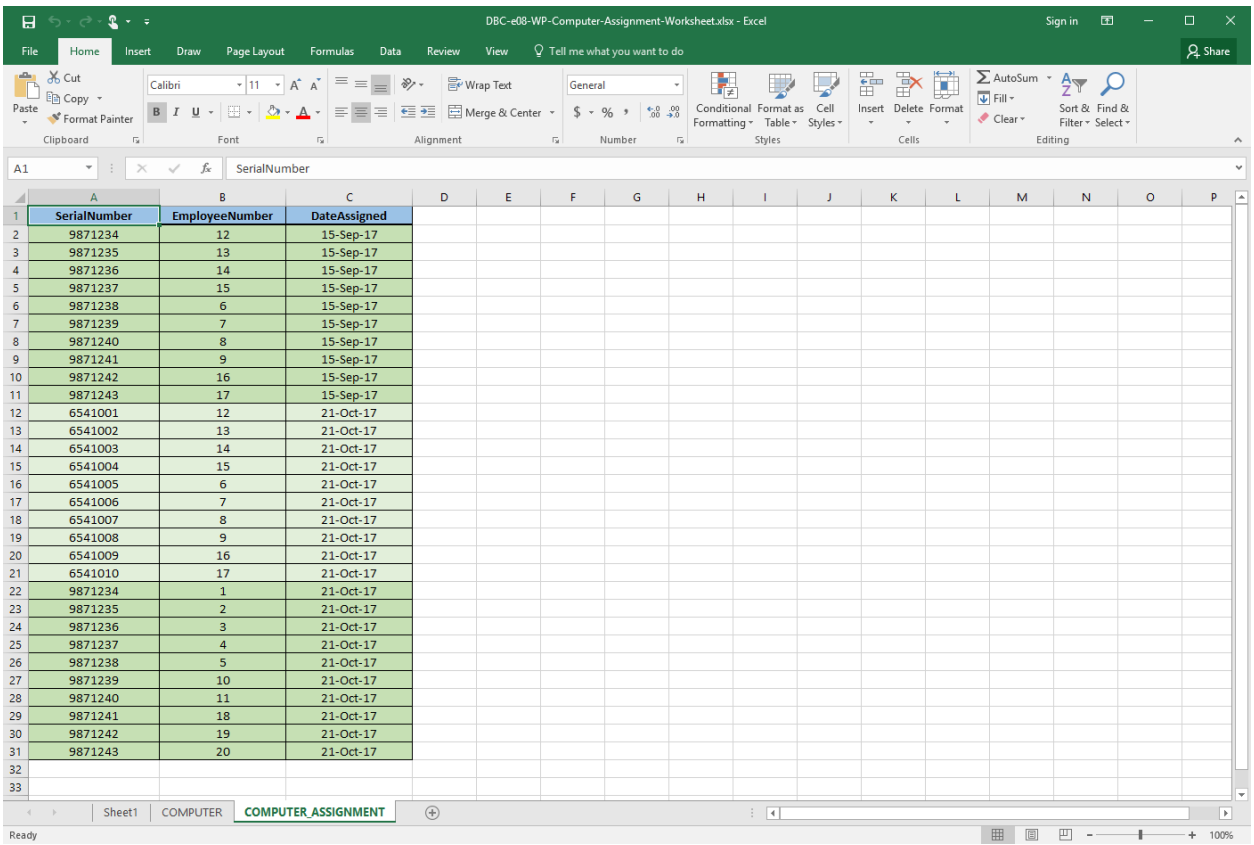


Figure B-15 — The WPC COMPUTER_ASSIGNMENT Worksheet – Second Attempt

and given a larger data set a different strategy would be needed. For our purposes here, however, this method will work.

We will next look at how to import a table into Oracle Database XE. The COMPUTER table column characteristics as stated in Figure 3-32 are repeated below in Figure B-16. Note that we will need to add CHECK constraints on this table and that neither of these can be done in the data import step—we will have to use SQL ALTER TABLE statements to implement these constraints after the table is created and the data are imported.

Importing the Microsoft Excel Data into an Oracle Database XE Table

We will import the WP COMPUTER table data from the spreadsheet into the Oracle Database XE WP database. The COMPUTER_ASSIGNMENT data can be imported in a similar way; this will be explored in the exercises.

Oracle Database, like all enterprise class DBMSs, provides a variety of ways to get external data into the DBMS. Be careful, however, with Oracle's terminology: what we are doing in this appendix, while generically referred to as *importing* data, is actually called *loading* data in Oracle. The term *import* in Oracle is reserved for copying data from one Oracle Database database into another. The Oracle Database XE graphical user interface provides wizards for loading (importing, in our terminology) data from spreadsheets, delimited text files (which can be easily produced from a spreadsheet application such as Microsoft Excel), or XML files. This method is appropriate for a small number of simple tables.

A more comprehensive tool for loading data into an Oracle Database XE database is Oracle's **SQL*Loader** utility. SQL*Loader can load (import) a wider variety of data and is capable of importing

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 B-16 — Database Column Characteristics for the WP COMPUTER Table

multiple tables with a single command. Data can be loaded across a network or from multiple files. In addition, data can be loaded selectively (only load rows matching a certain condition, for example) and can be manipulated using SQL during the import process. In this appendix, however, we will focus on importing data using SQL Developer, which we have already installed.

Oracle Database provides two ways of importing Microsoft Excel data via SQL Developer:

- Create the table first using an SQL CREATE TABLE statement, then import the data
- Create the table while importing the data

We will use the second method to import the COMPUTER data:

1. In Oracle SQL Developer, expand the WP Database.
2. Right-click on the **Tables (Filtered)** WP database object to display a shortcut menu, and in the shortcut menu click on the **Import Data** command, as shown in Figure B-17(a).

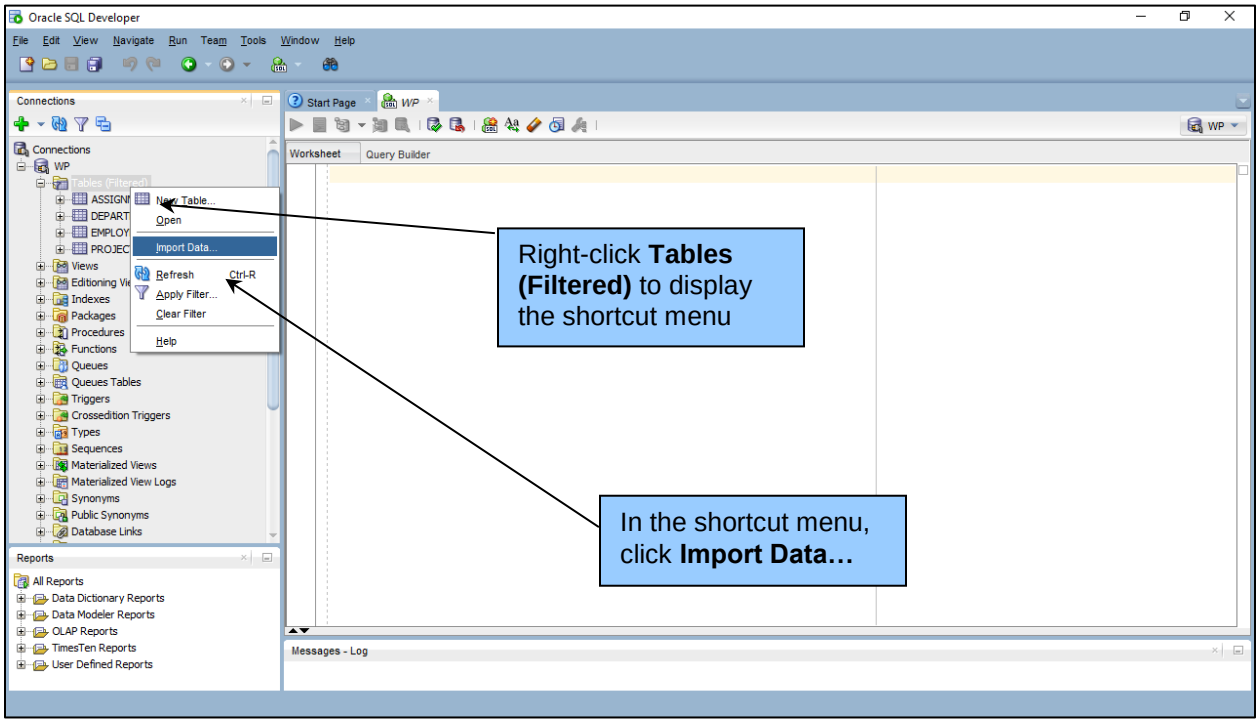


Figure B-17(a) — The Import Data Command

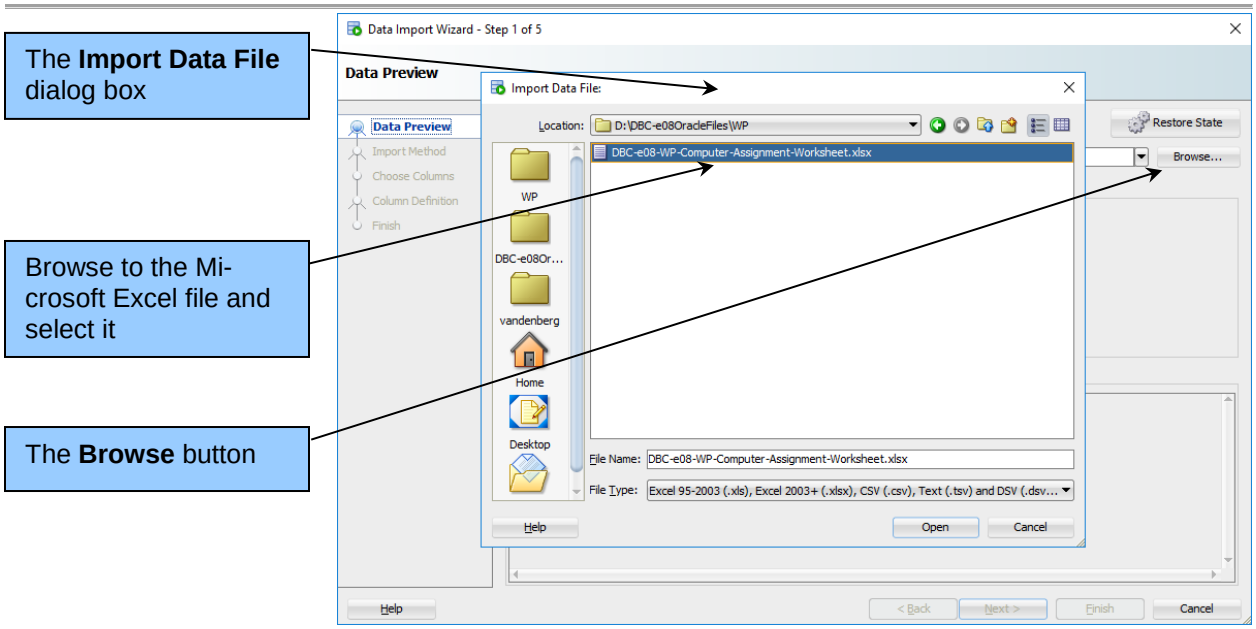


Figure B-17(b) — The Import Data File Dialog Box for the Excel Workbook File

3. The *Data Import Wizard-Step 1 of 5* (Data Preview) dialog box is displayed, as shown in Figure B-17(b). Use the **Browse** button to open the **Import Data File** dialog box and locate the Excel workbook containing the WP COMPUTER data. Click the **Open** button.
4. The Data Preview dialog box is redisplayed, this time with information from the Excel workbook. Select the **COMPUTER** worksheet from the Worksheet drop-down list as shown in Figure B-17(c). Also make sure the Header checkbox is checked and that the **Excel 2003+ (xlsx)** format is selected from the Format drop-down list.

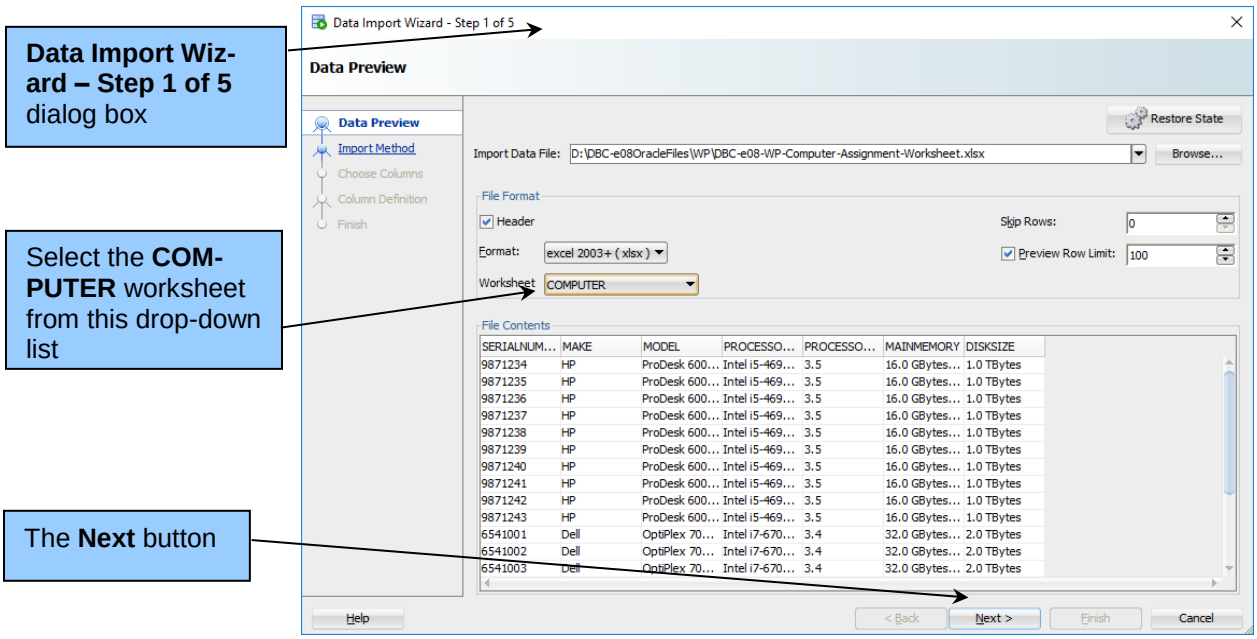


Figure B-17(c) — The Data Import Wizard – Step 1 of 5 Dialog Box

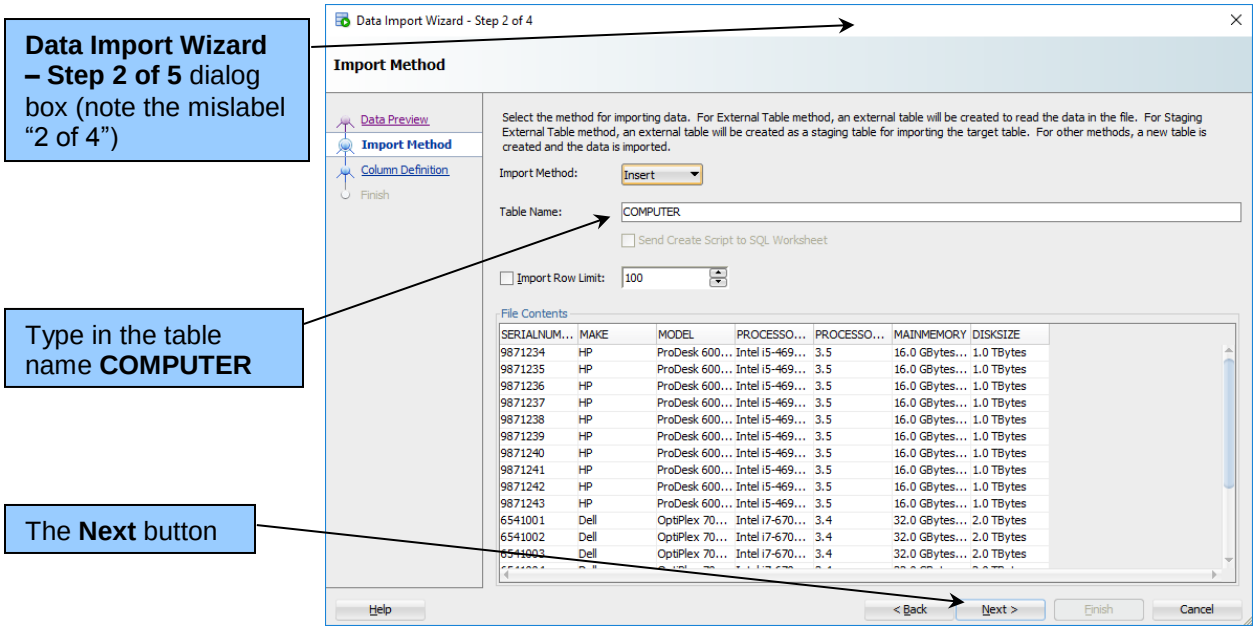


Figure B-17(d) — The Data Import Wizard – Step 2 of 5 Dialog Box

5. Click the **Next** button as shown in Figure B-17(c). The *Data Import Wizard – Step 2 of 5* (Import Method) dialog box is displayed (and mislabeled as *Step 2 of 4*). Type in the Table Name COMPUTER and leave the rest of the settings as they are, as shown in Figure B-17(d).
6. Click the **Next** button. The *Data Import Wizard – Step 3 of 5* (Choose Columns) dialog box is displayed as shown in Figure B-17(e). This step allows us to choose which spreadsheet columns to import. Note that all are currently selected, and that is what we want, so no changes are made.

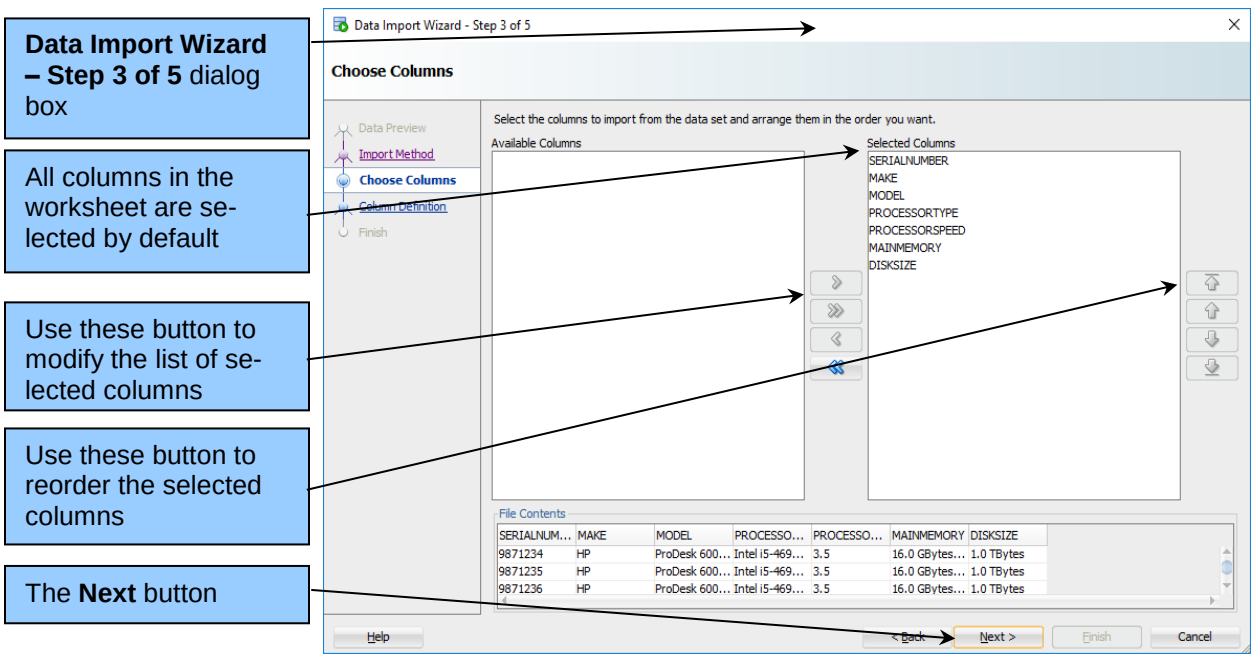


Figure B-17(e) — The Data Import Wizard – Step 3 of 5 Dialog Box

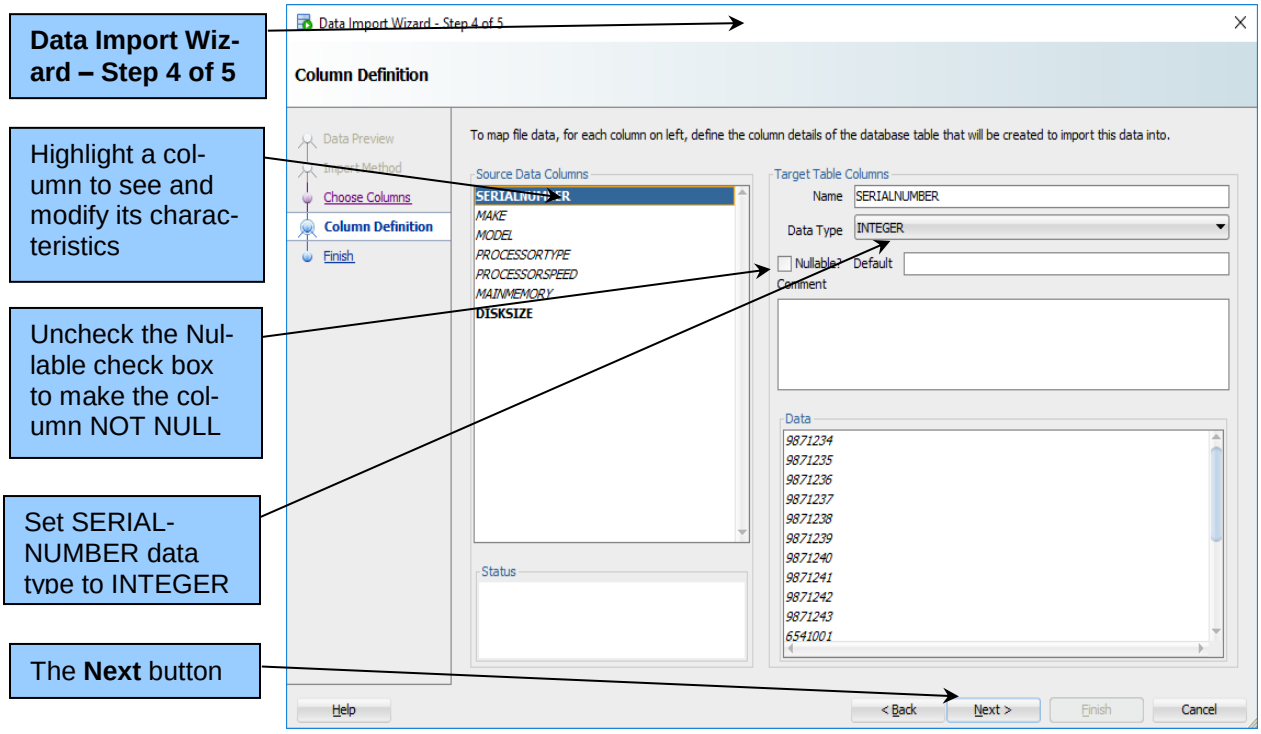


Figure B-17(f) — The Data Import Wizard – Step 4 of 5 Dialog Box

7. Click the **Next** button to display the *Data Import Wizard – Step 4 of 5* (Column Definition) dialog box as shown in Figure B-17(f). Here we can select column characteristics for the COMPUTER table. Note that in the figure we show the proper settings for the SERIALNUMBER column (which will become the primary key of the COMPUTER table). In particular, we set the column to **INTEGER** and **NOT NULL**.
8. Using Figures B-16 and B-17(f), edit the rest of the column characteristics to set the proper data type and nullability. For text fields, use the provided VARCHAR2 type, with the length as indicated in Figure B-16, with the following exception: For the MAINMEMORY column, use 16 for the column size, unless you want the import to fail. This is a good example of a common phenomenon encountered when importing data: the presence of spaces or other whitespace in the source file can cause problems. In this case, we can easily alter the column later to be 15 characters long, using the SQL ALTER TABLE command. For PROCESSORSPEED, use a Data Type of NUMBER with Precision 3 and Scale 2. When you have finished, click the **Next** button.
9. The *Data Import Wizard – Step 5 of 5* (Finish) dialog box is displayed, as shown in Figure B-17(g). This window allows you to examine the choices made during the import dialogs and to save the settings to a file for future use (perhaps with another, similar table or spreadsheet). Click the **Finish** button.
10. The Import Data results box appears, indicating successful completion of the importing, as shown in Figure B-17(h). Click **OK** to complete the import process.

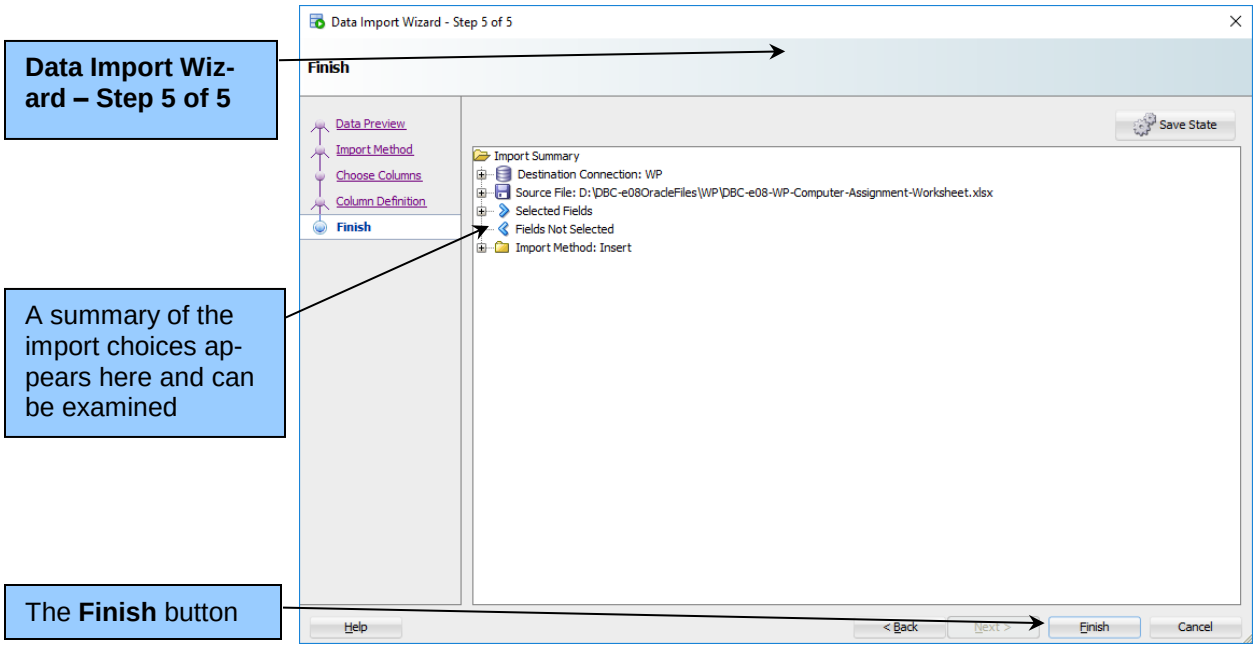


Figure B-17(g) — The Data Import Wizard – Step 5 of 5 Dialog Box

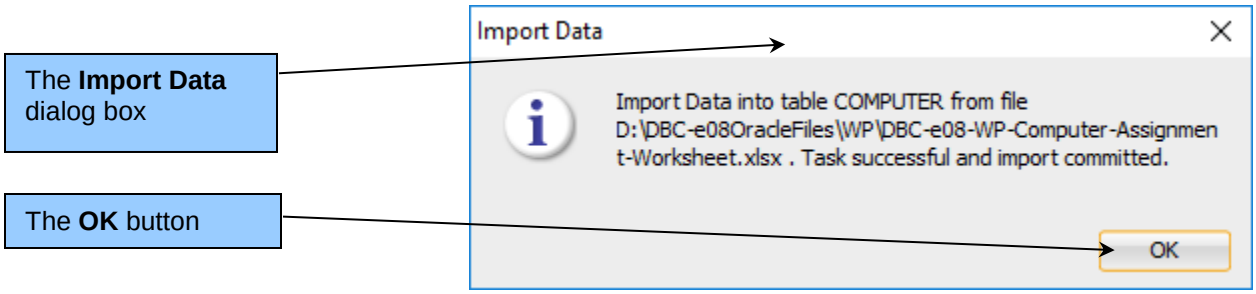


Figure B-17(h) — The Import Data Results Box

11. In the WP SQL Worksheet window, type and run the following SQL query to verify the proper importing of the COMPUTER data:

```
/* *** SQL-Query-AppB-02 *** */
SELECT   *  FROM  COMPUTER;
```

12. The data should be correct, but recall that we had to specify the MAINMEMORY column as a 16-byte character string instead of 15 in order for the import to work. Based on our examination of the data in the previous step, it is clear that the correct data are in there and the 16th character of each MAINMEMORY value must be a space. This means we can shorten that column to 15 characters without losing any information. This can be done using an SQL UPDATE command and an SQL ALTER TABLE command: First we have to actually shorten each MAINMEMORY

value to remove the ending whitespace then we can change the size of the column without losing any data. Type and run the following SQL UPDATE command in the WP SQL Worksheet (the SQL UPDATE command is covered in Chapter 3):

```
/* *** SQL-Update-AppB-01 *** */
UPDATE   COMPUTER SET MAINMEMORY = substr(MAINMEMORY, 1, 15);
```

The ALTER TABLE command will be explored in more detail in Appendix E, but for now just type the following command in the WP SQL Worksheet and click the **Run Statement** button:

```
/* *** SQL-Alter-Table-AppB-01 *** */
ALTER TABLE COMPUTER MODIFY MAINMEMORY VARCHAR2(15);
```

13. The one remaining task for the COMPUTER table is to set SERIALNUMBER as the primary key. The ALTER TABLE syntax for accomplishing this will be covered in Appendix E.
14. The COMPUTER table has been added to the WP database.

We have imported the COMPUTER table into the WP database. Importing the rest of the data from the spreadsheet (i.e., the COMPUTER_ASSIGNMENT data) will be explored in the exercises.

How Do I Create an ODBC Connection from Microsoft Access 2016 to an Oracle Database XE Database?

While Oracle Database is an excellent enterprise-class DBMS, its SQL Developer environment does not provide any application development tools. Oracle Application Express does provide the ability to create Web-based forms and reports. Oracle Forms and Reports is an Oracle product that can create full-featured applications. In many cases, however, it is useful to be able to do quick prototyping and implementation of simple reports and forms using software that many people already have. 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 Oracle Database XE database.

This can be done 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. ODBC connections require programs called *clients* and *drivers* in order to establish the connection. Installation of Oracle Database XE installs both these components automatically, so no further installations need to be done in order to use ODBC to connect Microsoft Access 2016 to Oracle Database XE.

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 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**, use the folder icon to navigate to the folder where you wish to store the database, 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 work 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 B-18(a).
4. Click the **OK** button to save the settings and close the dialog box.

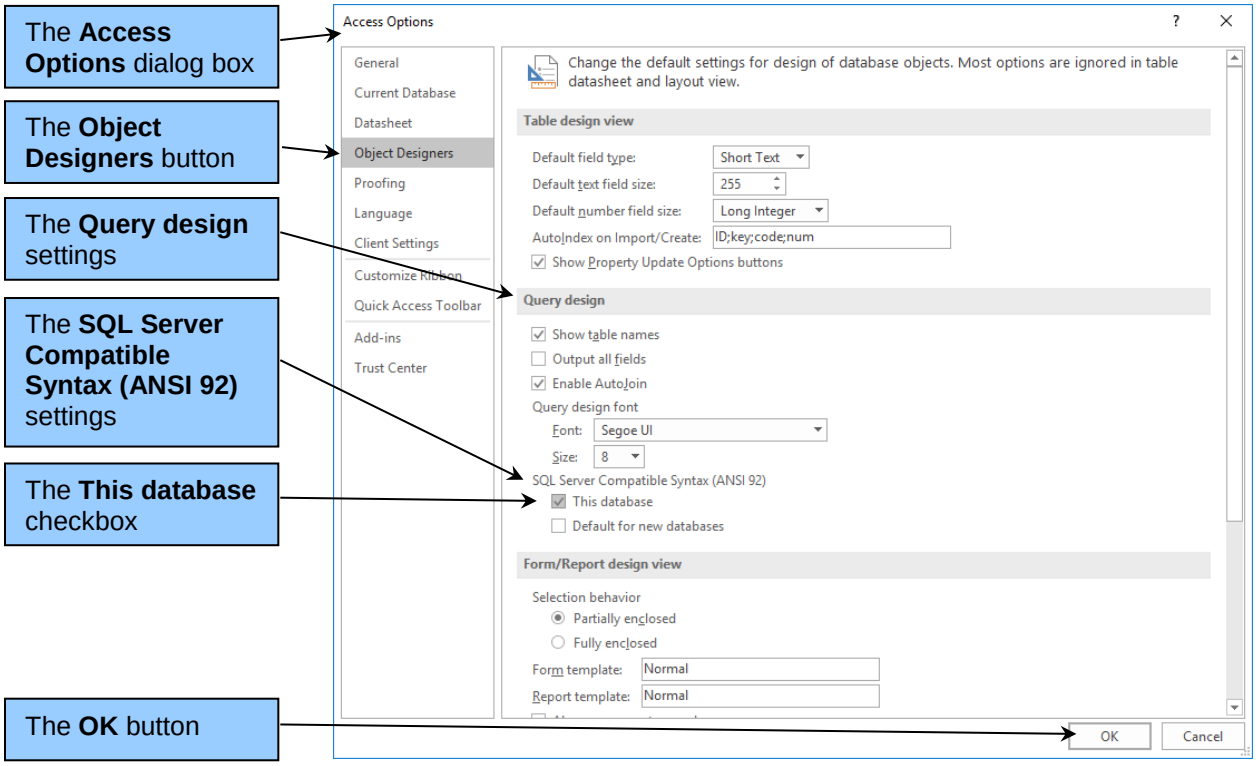


Figure B-18(a) — The SQL Server Compatible Syntax (ANSI 92) Checkbox

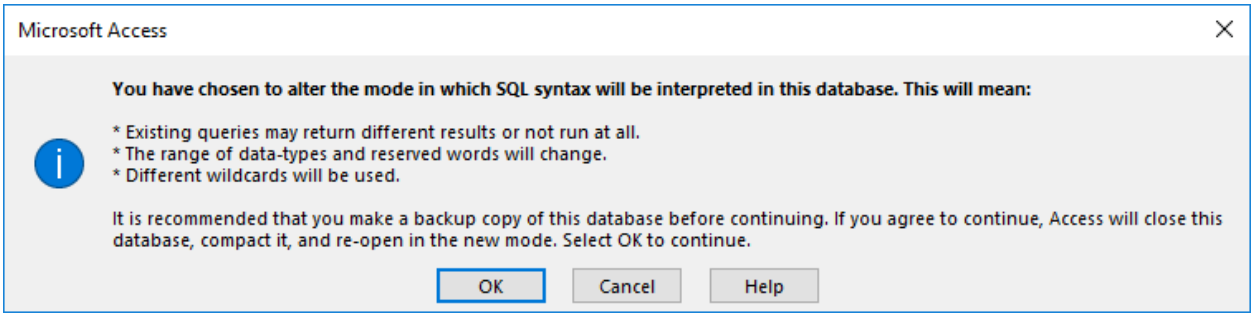


Figure B-18(b) — The Microsoft Access Information Dialog Box

5. A Microsoft Access Information dialog box is displayed, as shown in Figure B-18(b).
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 Oracle Database XE. To do this we create a link to data external to Microsoft Access by using an 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 previous 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 B-19(a).
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 B-19(a), and then click the **OK** button.
4. The **Select Data Source** dialog box is displayed, as shown in Figure B-19(b). This is the dialog box that we will use to create the needed OBDC 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 B-19(c).
6. In the Create New Data Source dialog box, scroll down through the list of drivers until you can see the driver named **Oracle in XE**. Click this driver name to select it, and then click the **Next** button.

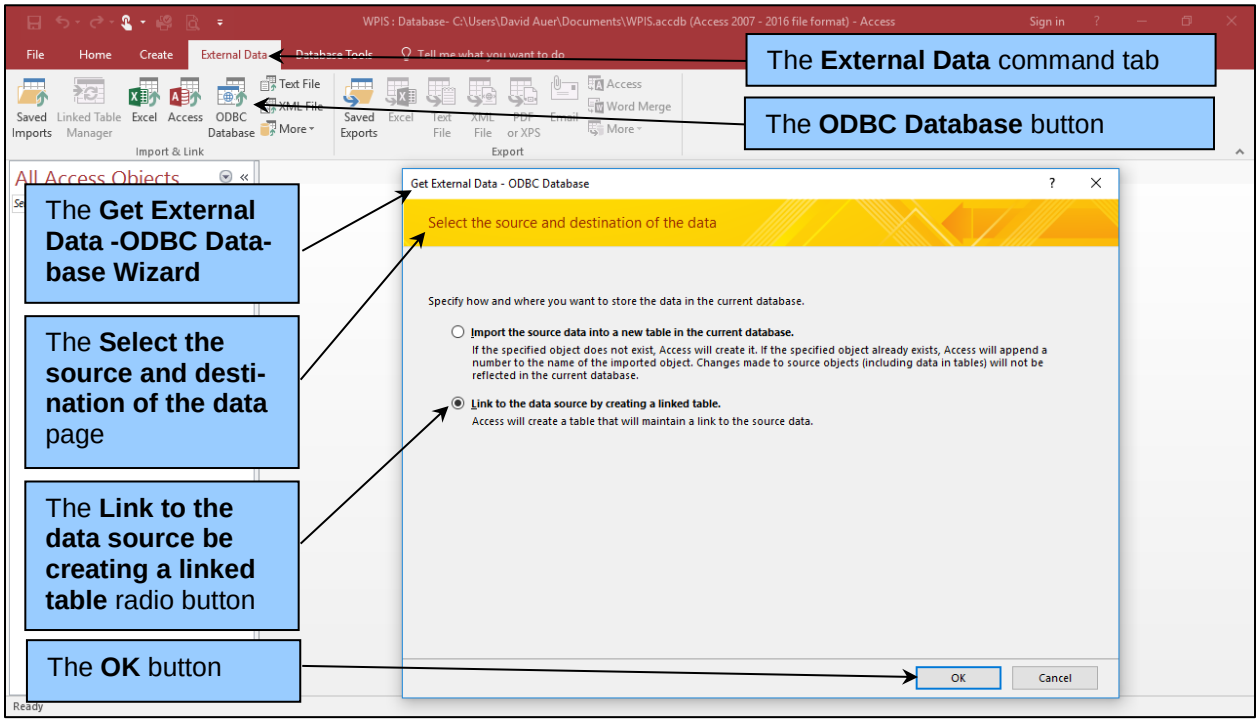


Figure B-19(a) — The Get External Data - ODBC Database Wizard Dialog Box

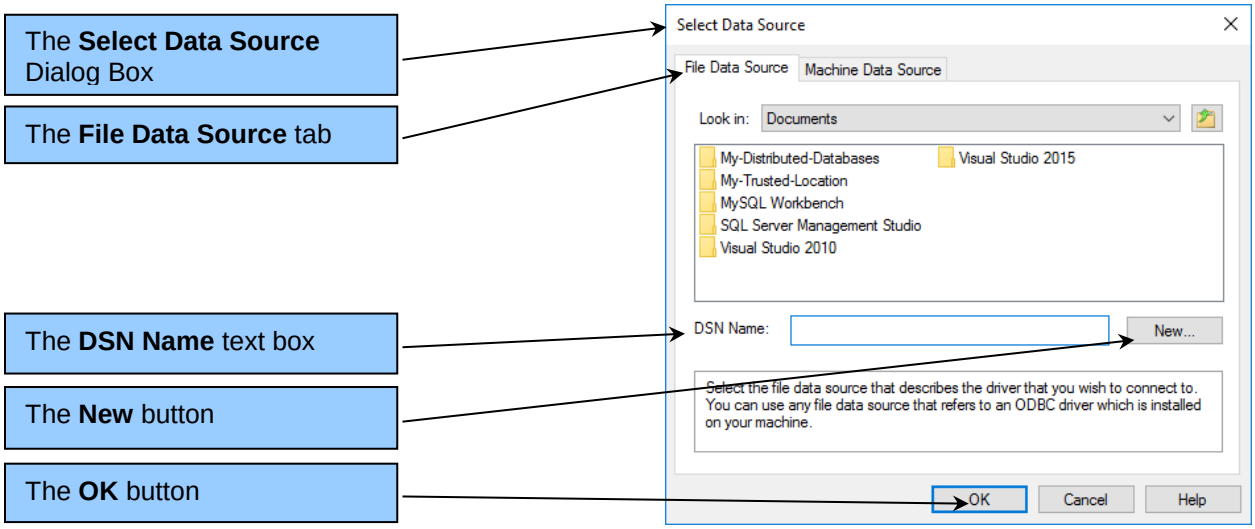


Figure B-19(b) — The Select Data Source Dialog Box

7. As shown in Figure B-19(d), 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 B-19(e), the next page of the Crate New Data Source dialog box provides a summary of the settings that will be used to create the new DSN. Click the **Finish** button.

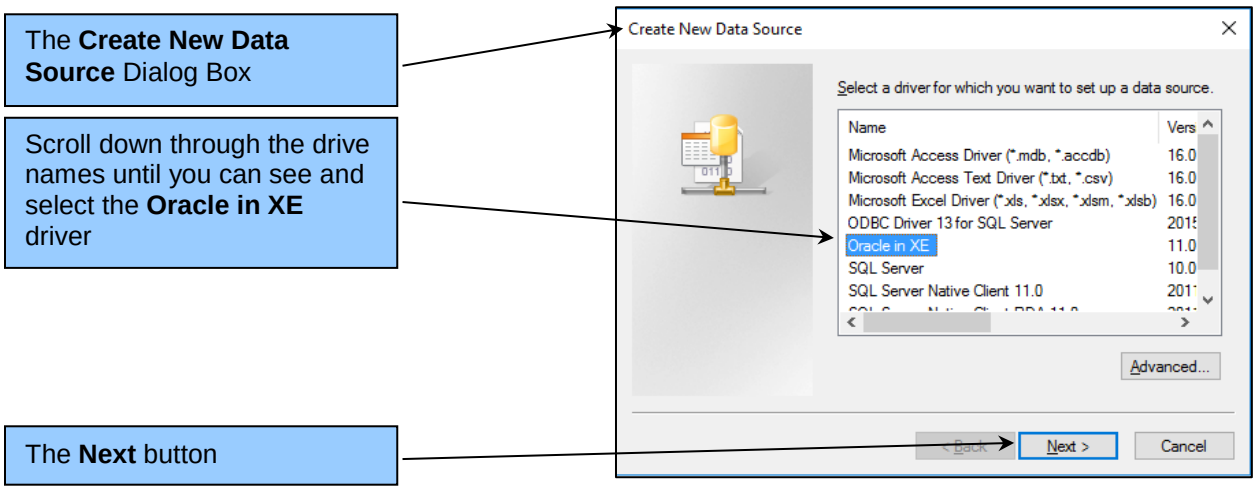


Figure B-19(c) — The Create New Data Source Dialog Box

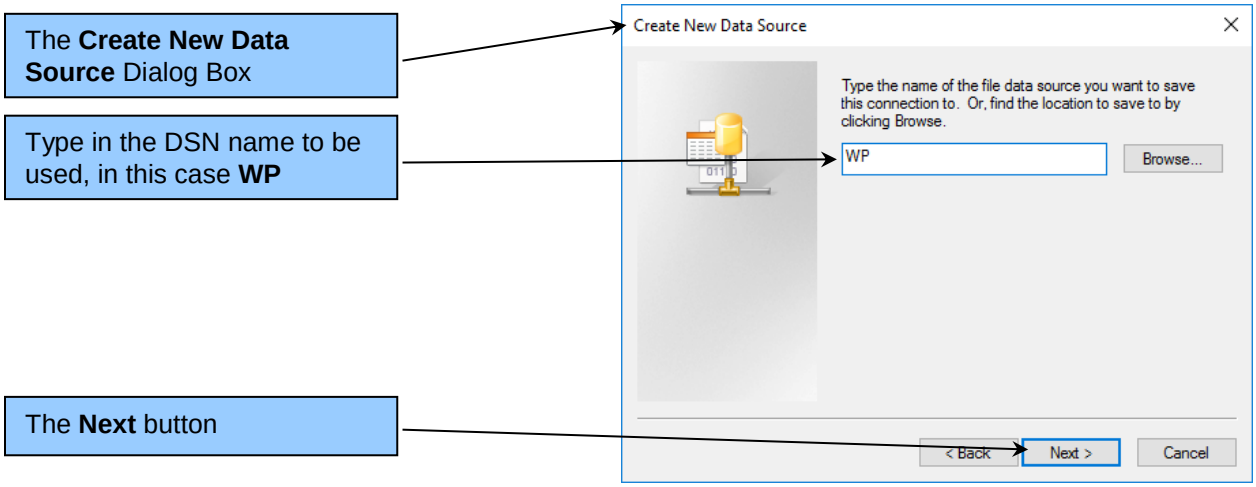


Figure B-19(d) — Naming the DSN

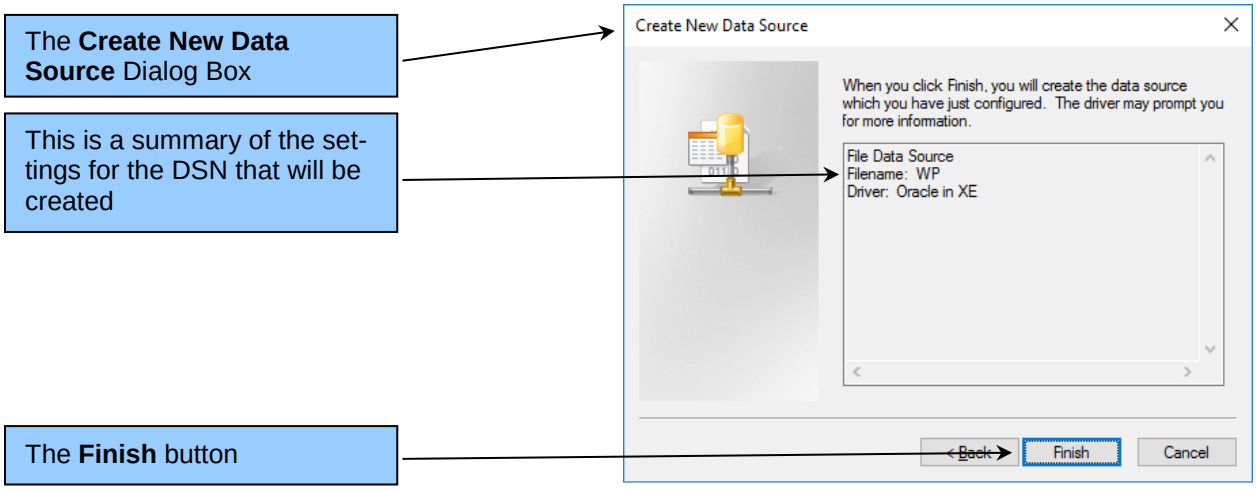


Figure B-19(e) — The Create New Data Source Dialog Box

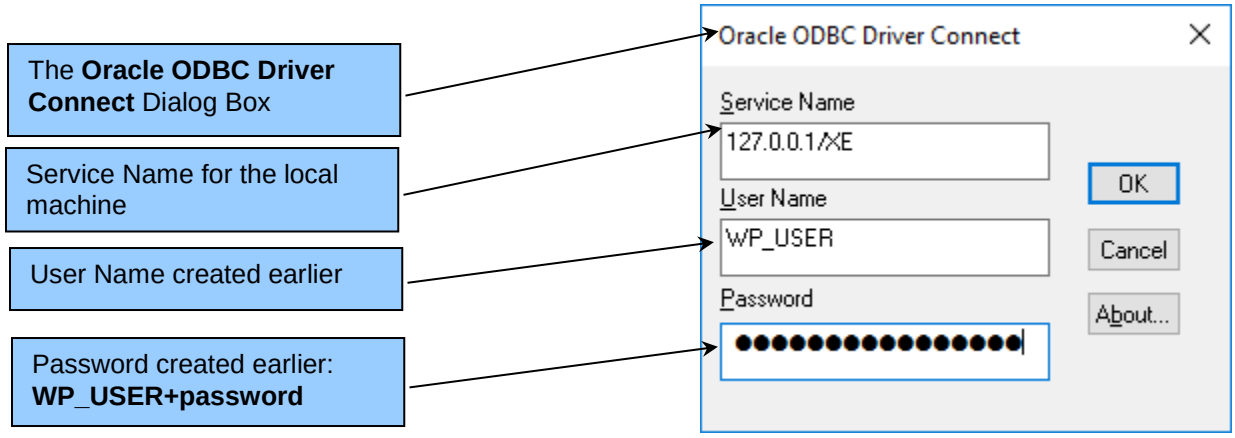


Figure B-19(f) — The Oracle ODBC Driver Connect Dialog Box

9. The **Oracle ODBC Driver Connect** dialog box is displayed, as shown in Figure B-19(f). Enter the **Service Name**, User Name, and Password as shown in the figure (the User Name and Password were created earlier in this appendix). Click the **OK** button.
10. The WP DSN file data source is created and displayed in the Select Data Source dialog box as shown in Figure B-19(g). Now that the DSN is completed, we can finish linking the Microsoft Access 2016 database to the Oracle Database XE database.

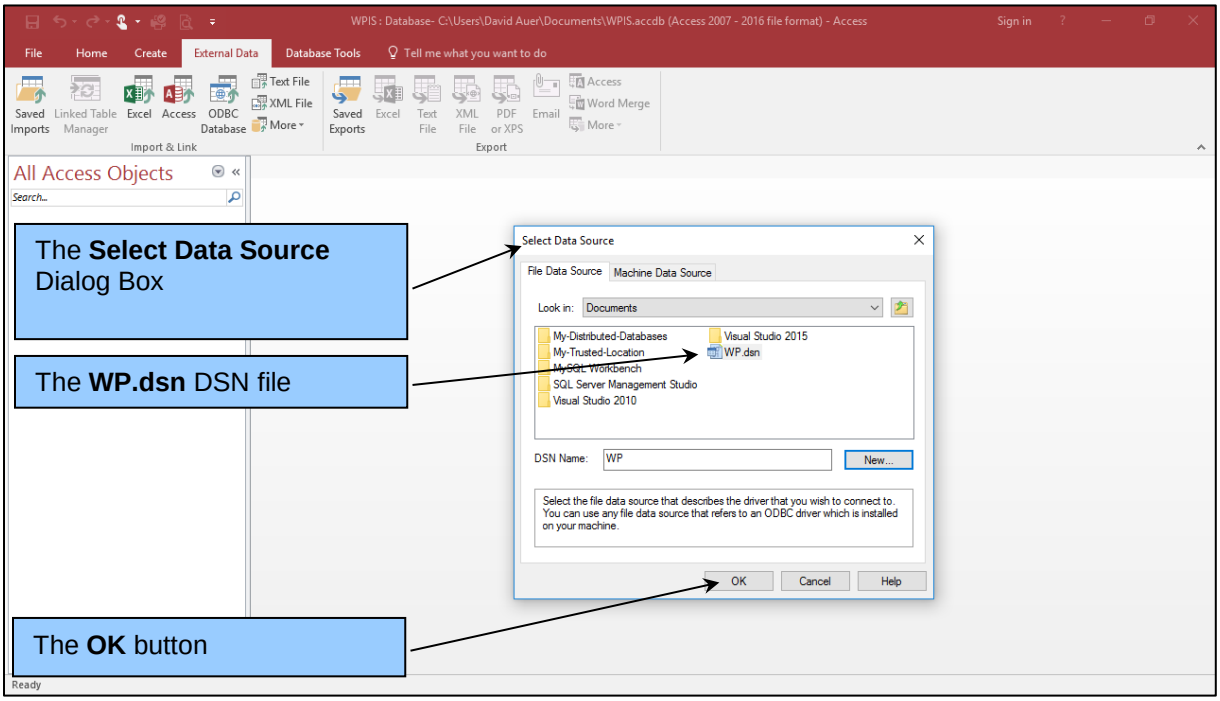


Figure B-19(g) — The Completed WP DSN

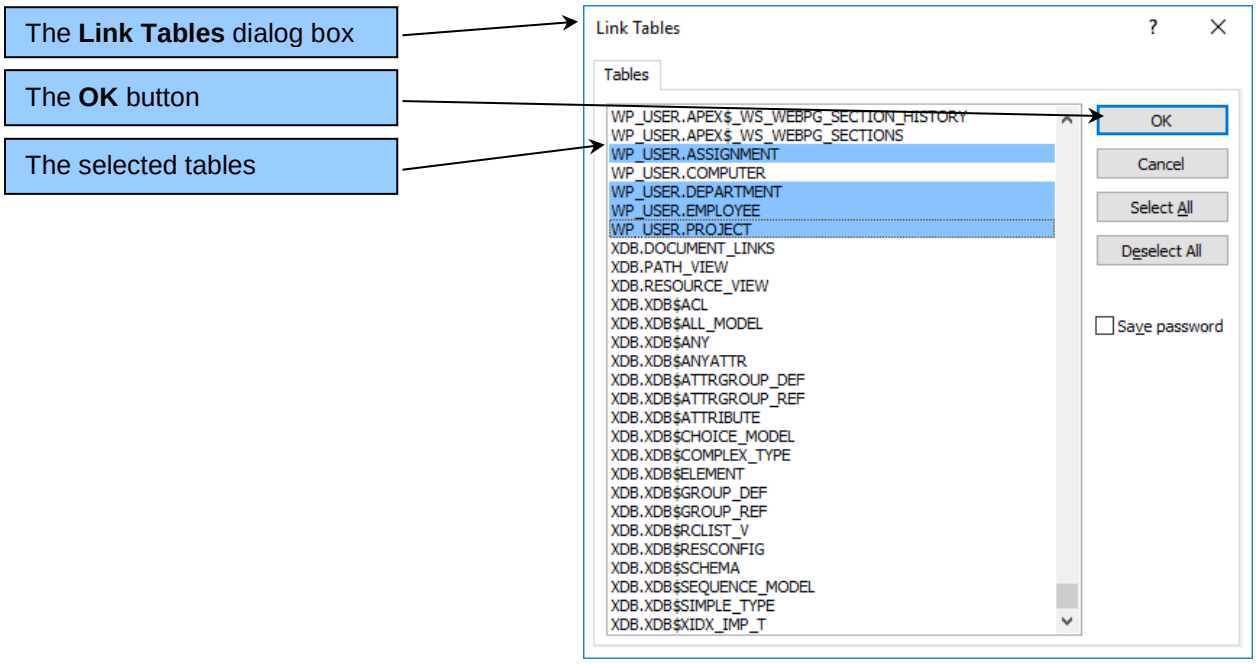


Figure B-19(h) — The Link Tables Box

11. In the **Select Data Source** dialog box, click the **OK** button.
12. The **Oracle ODBC Driver Connect** dialog box is displayed again, as shown in Figure B-19(f). Enter the password **WP_USER+password** in the **Password** text box, and then click the **OK** button.
13. The **Link Tables** dialog box is displayed, as shown in Figure B-19(h). There are many built-in **metadata tables** here that we are not interested in. Scroll down until you find the WP_USER tables. Note that four tables have been selected. To select the **ASSIGNMENT** table, **click** on it. To add each additional table to the selection, **click on the table name**. We will not link the COMPUTER table at this time since we have not yet given it a primary key.
14. Click the **OK** button. The ODBC links between the WP.accdb Microsoft Access 2016 database and the WP database in Oracle Database XE are completed, as shown in Figure B-19(i).
15. 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 B-19(j). The data shown in this form are actually in the Oracle Database XE WP database.
16. Close the WP.accdb Microsoft Access database.

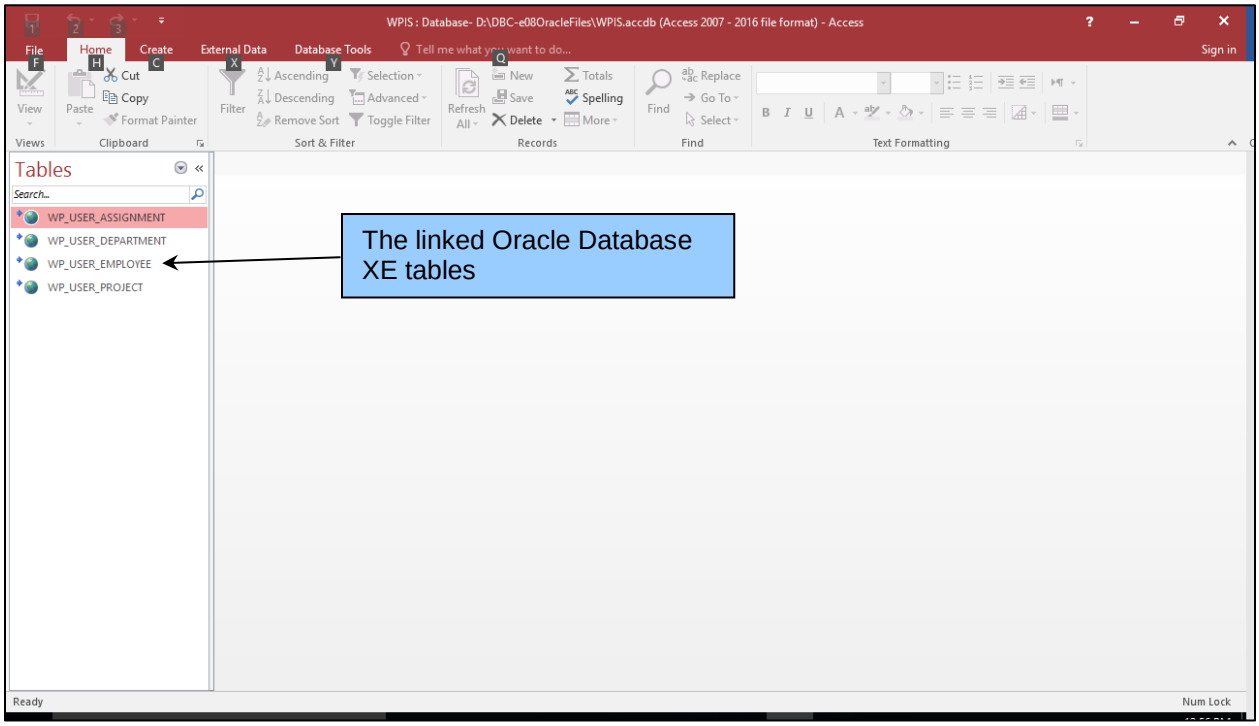


Figure B-19(i) — The Linked Oracle Database XE Tables in the Microsoft Access Database

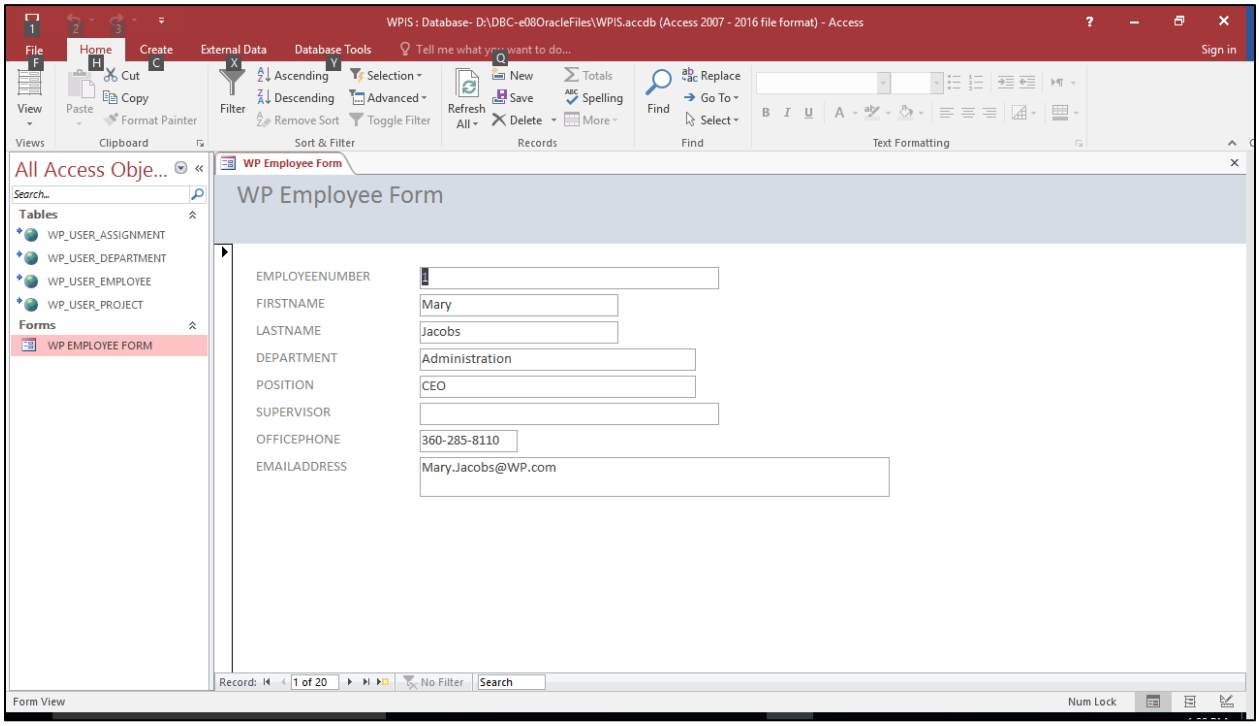


Figure B-19(j) — The WP Employee Form

Additional Documentation for Oracle Database XE

To get access to a complete set of Oracle Database XE documentation, visit the Oracle home page (www.oracle.com). As in Figure B-2(a), navigate to the download page for Oracle Database XE. Then click on the documentation tab shown in Figure B-2(b).

KEY TERMS

***.sql**	**ANSI standard SQL (ANSI 92)**
commit	**connection**
CurrVal	**DSN (data source name)**
dynamic host configuration protocol (DHCP)	**extract, transform, and load (ETL)**
filter	**import**
loopback adapter	**metadata tables**
Microsoft ANSI-89 SQL	**multivalue, multicolumn problem**
multivalued dependency	**NextVal**
open database connectivity (ODBC)	**Oracle Application Express**
Oracle Database 12c	**Oracle Database XE**
Oracle Java Development Kit (JDK)	**Oracle SQL Developer**
rollback	**run script button**
run statement button	**sequence**
service name	**SQL script**
SQL Worksheet	**SQL*Loader**
System Identifier (SID)	**SYSTEM account**
TO_DATE	**workspace**

REVIEW QUESTIONS

B.1 What is Oracle Database XE?

B.2 What is the primary advantage of using Oracle Database XE instead of Microsoft Access?

B.3 What are the two Oracle programs that are recommended as a necessary set of Oracle Database XE software products? In what order should you install these products?

B.4 How do you install Oracle Database XE?

B.5 What is the purpose of the Oracle Database 11*g* XE Web utility?

B.6 How do you install Oracle SQL Developer?

B.7 What is the purpose of the Oracle SQL Developer utility?

B.8 How do you create a new database in Oracle Database XE?

B.9 How do you connect to a database in Oracle Database XE?

B.10 What is an SQL script? What types of SQL statements and commands can you run more efficiently as scripts?

B.11 What tool(s) can be used to create a script?

B.12 What file extension should you use for SQL scripts?

B.13 How do you open and run a script in Oracle SQL Developer?

B.14 How do you create database tables in Oracle SQL Developer?

B.15 What is a *sequence*? How are sequences used in Oracle Database XE?

B.16 How do you populate database tables in Oracle SQL Developer?

B.17 How do you create and run an SQL query in Oracle SQL Developer?

B.18 How do you import Microsoft Excel data into an Oracle Database XE database?

B.19 Why would you want to create an ODBC connection to link a Microsoft Access 2016 database to an Oracle Database XE Database?

B.20 What is an ODBC DSN? Why is one needed?

B.21 How do you create an ODBC connection to link a Microsoft Access 2016 database to an Oracle Database XE database?

EXERCISES

B.22 If you haven't already done so, download and install Oracle Database XE as described in the text. Use the default settings for the installation. Be sure that Oracle SQL Developer is also correctly installed.

B.23 If you haven't already done so, work through the steps described in this appendix to create and populate the WP database.

B.24 Using Oracle Database XE and SQL Developer, run the following SQL queries from Chapter 3:

SQL-QUERY-CH03-01 through SQL-QUERY-CH03-32

SQL-QUERY-CH03-35 through SQL-QUERY-CH03-56 (remember that the "AS" keyword is not allowed by Oracle with the JOIN syntax)

Save each query as follows:

- Create and run each query in Oracle SQL Developer.
- After you have run each query, save the query. By default, Oracle Database XE 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 **ODB-SQL-Query-01.sql**.

B.25 Use Oracle SQL Developer to run one or more of the *saved* SQL queries you created in the previous question:

- 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.

B.26 Complete exercise 3.59 using Oracle Database XE and Oracle SQL Developer. Start each saved query name with **ODB-** and use the default ***.sql** file extension. (The first saved query name should be **ODB-SQL-Query-AWE-3-1-A.sql**.)

B.27 Complete exercise 3.60 using Oracle Database XE and Oracle SQL Developer. Start each saved script or query name with **ODB-** and use the default ***.sql** file extension. Create as many scripts as you need for Parts A-D. The saved query name will be **ODB-SQL-Query-AWE-3-3-E.sql**.

B.28 If you have not already done so, import the **COMPUTER** table from Microsoft Excel into the Oracle Database XE WP database as explained in the text.

B.29 Import the **COMPUTER_ASSIGNMENT** table from Microsoft Excel into the Oracle Database XE 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 Oracle Database XE. Note that this may require using some ALTER TABLE statements as discussed in Appendix E.

B.30 Create a Microsoft Access 2016 database named the **WPIS_CO.accdb** where CO stands for "computer-only." This database will be an application for the Oracle Database XE WP database which will allow users to read, query, and alter the data in the WP database COMPUTER table but will provide no access to other WP database tables. Once you have created the database:

1. If you have not completed exercise B.28, do so now.
2. Set the WPIS_CO.accdb database to use **SQL Server Compatible Syntax (ANSI 92)**.
3. Link the WPIS_CO.accdb database to the Oracle Database XE WP database using ODBC. When you create your File Data Source DSN, name it **WPCO**, and use the same WP_USER account for Oracle Database XE authentication.
4. Import the COMPUTER table. You may be asked by Access to specify a primary key; if so, use SERIALNUMBER.
5. Create a form to show all the data in the COMPUTER table named **WP Computer Form**.
6. Create a report to show all the data in the COMPUTER table named **WP Computer Report**.