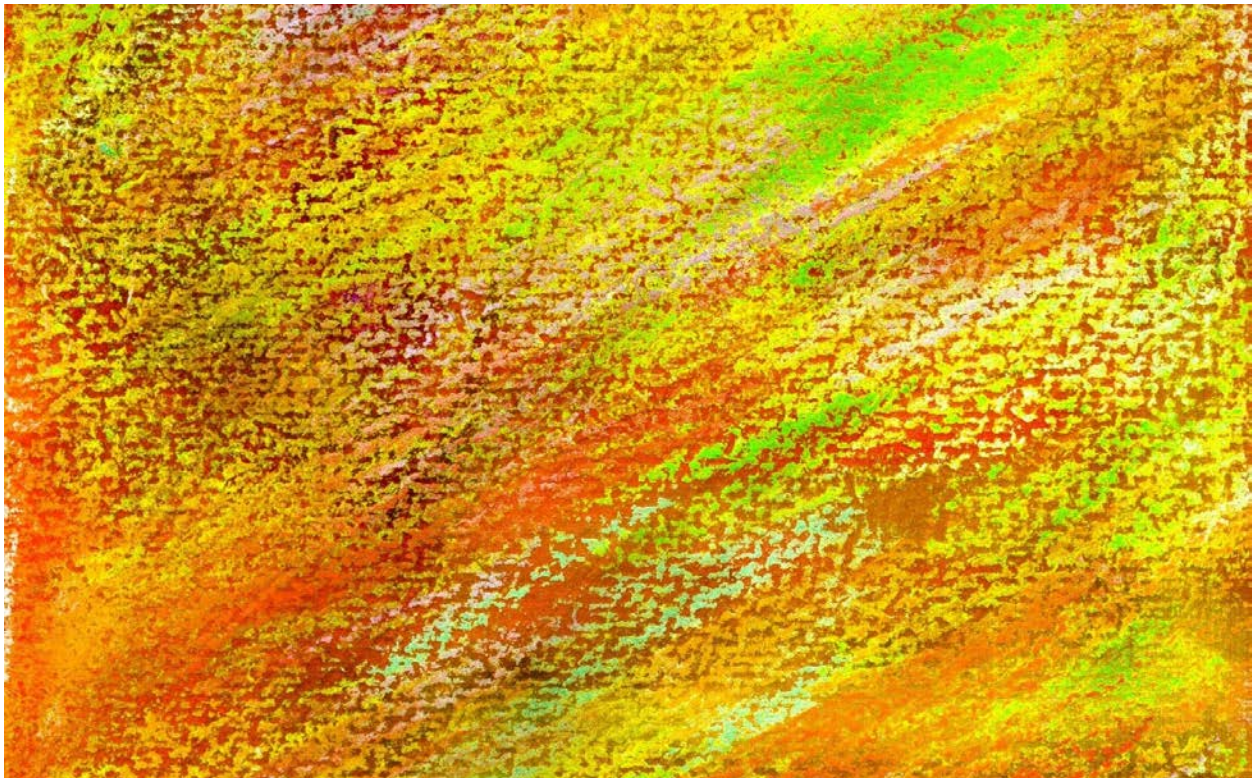# Database Concepts

7th Edition

David M. Kroenke • David J. Auer

Online Appendix A

Getting Started with
Microsoft SQL Server 2014 Express

Chapter Objectives:

- Learn how to create a database in SQL Server 2014
- Learn how to submit SQL commands to create table structures
- Learn how to submit SQL commands to insert database data
- Learn how to submit SQL commands to query a database
- Learn how to install the Microsoft SQL Server 2014 ODBC Client

What is the Purpose of this Appendix?

Microsoft SQL Server is an enterprise-class DBMS that has been around for many years. SQL Server 2005, including **SQL Server 2005 Express Edition** was released in November 2005. **SQL Server 2008** and **SQL Server 2008 R2** were released in August 2008; SQL Server 2008 and SQL Server 2008 R2 also included an Express version. **SQL Server 2012** was released in March 2012. Now **SQL Server 2014 Express** has been released and is available for use with this book.[1] The SQL Server Express Editions seem to be designed to compete with Oracle's MySQL (see Appendix C). MySQL is an open-source database that,

---

[1] As this book goes to press, the current version of SQL Server Express Edition is SQL Server 2014 Express. No service packs have been issued for SQL Server 2014. Always be sure you download and install the latest service pack for whichever version of SQL Server or any other DBMS you are using.

while not having as many features as SQL Server, has had the advantage of being available for free via download over the Internet. It has become widely used and very popular as a DBMS, supporting Web sites running the Apache Web server.

SQL Server Express is becoming increasingly popular. Microsoft releases various versions of SQL Server, and while SQL Server Express is the least powerful version, it is intended for general use and can be downloaded for free.

SQL Server 2014 Express is available in a basic package and in an advanced features package. The version with advanced features is called **SQL Server 2014 Express Advanced**. Included in the advanced features package is Microsoft's GUI SQL Server management tool and support for SQL Server Reporting Services. Both of these features are well worth having (otherwise you'll need to download the management tool separately), so download and install the version with the advanced features package. The screen shots appearing in this book use SQL Server 2014 Express Advanced running in the Windows 7 operating system.

By The Way

To install Microsoft SQL Server 2014, you will need to know whether you are running a 32-bit or 64-bit operating system. Windows 8—and earlier versions of the Windows operating system—is available in both 32-bit and 64-bit versions. To determine which version you have, click the **Start** button, right-click **Computer** in the right hand column, and click **Properties**. Look at the value of **System Type** in the **System** information group.

The 32-bit versions of Microsoft programs (including, but not limited to, SQL Server 2014 Express) at the download sites are designated as **x86**, while the 64-bit versions are designated as **x64**. The "x86" refers to Intel processors that include "86" in the processor name (for example, the Intel 80486 CPU chip) and related processors such as the Intel Pentium CPU chip (which would have been the 80586 if Intel hadn't switched to names instead of numbers for their product line).

Why Should I Learn to Use SQL Server?

For the purposes of this book, the most important reason to learn to use Microsoft SQL Server 2014 is that SQL Server really handles SQL well. All the SQL results shown in Chapter 3 and Appendix E were created in SQL Server 2014. All the SQL commands and keywords in Chapter 3 and Appendix E marked "Does Not Work with Microsoft Access ANSI-89 SQL" will work with SQL Server.

What Will This Appendix Teach Me?

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

What Won't This Appendix Teach Me?

The material in this appendix does not go beyond what is necessary to get you started. There are many important SQL Server topics not covered here. Stored procedures and triggers are covered in Appendix E. Backups and restores, and database security are covered in David M. Kroenke and David J. Auer, *Database Processing: Fundamentals, Design, and Implementation*, 13th edition (Upper Saddle River, NJ: Prentice Hall, 2014).

How Do I Install SQL Server 2014 Express?

To install SQL Server 2014 Express edition with advanced options, you may need to download the following programs:

1.   Microsoft Windows Installer 4.5. Before installing any of the SQL Server products, you must install Windows Installer 4.5 (http://www.microsoft.com/en-us/download/details.aspx?id=8483). This version, or a later version should have been installed when you installed the Windows OS (Windows 7 and later), but we list this resource in case you need a separate installation).

2.   Microsoft .NET Framework 3.5 Service Pack 1. This is an update to the Microsoft .NET Framework 3.5 (http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=22).

3.   Microsoft .NET Framework 4.5.1. This is an updated version of the Microsoft .NET Framework. (http://www.visualstudio.com/downloads/download-visual-studio-vs).

4.   Microsoft SQL Server 2014 Express Advanced. Note that **Microsoft SQL Server 2014 Management Studio** is included in the download of Microsoft SQL Server 2014 Express Advanced. If you choose to download the basic version of SQL Server 2014 Express, you must download and install this program separately. Microsoft SQL Server 2014 Management Studio is the graphical management utility for SQL Server 2014 in general and SQL Server 2014 Express edition in particular. SQL Server 2014 is a command line–oriented program. SQL Server 2014 Management Studio makes it much easier to work with SQL Server.

Download SQL Server 2014 Express Advanced from http://www.microsoft.com/en-us/download/details.aspx?id=42299 . Download ***ExpressAdv 32BIT\SQLEXPADV_x86_ENU.exe*** if you are running a 32-bit operating system, and download **ExpressAdv 64BIT\SQLEXPADV_x64_ENU.exe** if you are running a 64-bit operating system.

Be aware that SQL Server 2014 is an enterprise-class DBMSs and, as such, is much more complex than Microsoft Access. Further, they do not include application development tools, such as form and report generators. For more information on the Microsoft Express series of products, a good place to start is www.microsoft.com/express.

Starting the Microsoft SQL Server Management Studio Express Edition

This appendix uses SQL Server 2014 – early versions of Microsoft SQL Server function similarly. To start working with SQL Server 2014 Express in Windows 8 and 8.1 Update 1, double click the **SQL Server Management Studio** icon in the Start window (in the Windows 7 operating system, select **Start | All Programs | Microsoft SQL Server 2014 | SQL Server Management Studio**.) The Microsoft SQL Server Management Studio **Connect to Server** dialog box appears, as shown in Figure A-1.

By The Way

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

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

If you use the default **Windows Authentication** method, you will be authenticated using your current Windows username and password, so just click the **Connect button**. When you complete your login, the Microsoft SQL Server Management Studio window appears, as shown in Figure A-2.
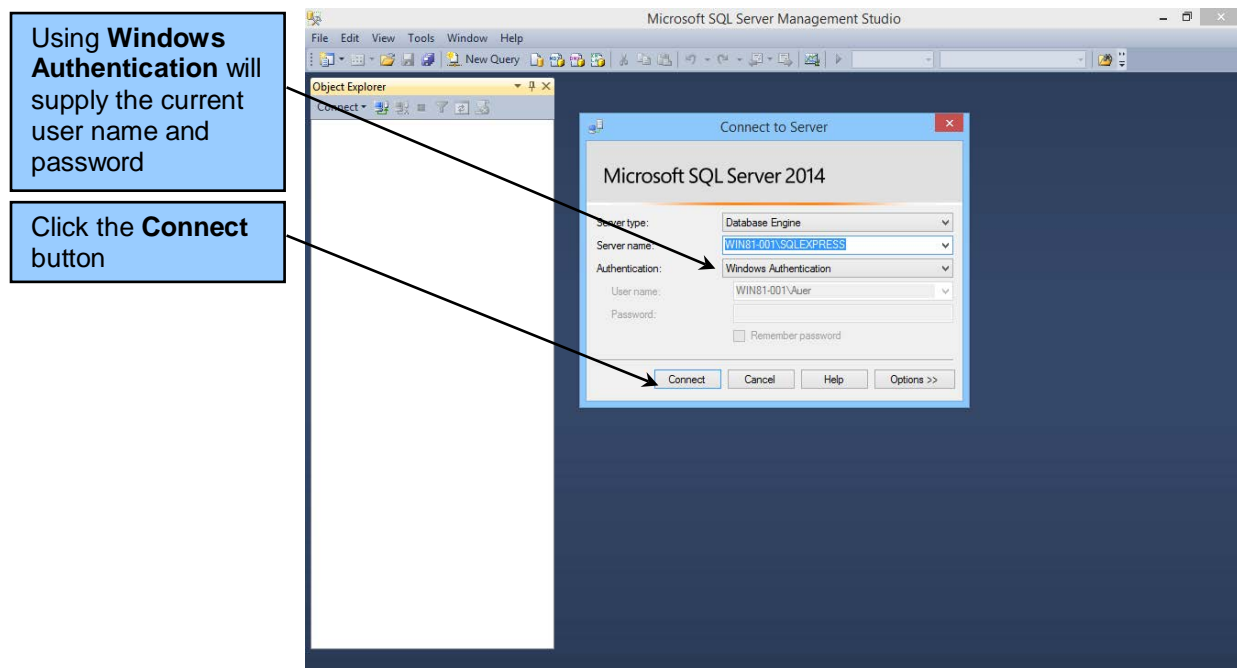
Figure A-1 — The Connect to Server Dialog Box

The **Object Explorer** shows SQL Server 2014 objects and the folders that are used to contain and organize those objects

The **System Databases** folder contains database objects for the databases automatically created by SQL Server 2014
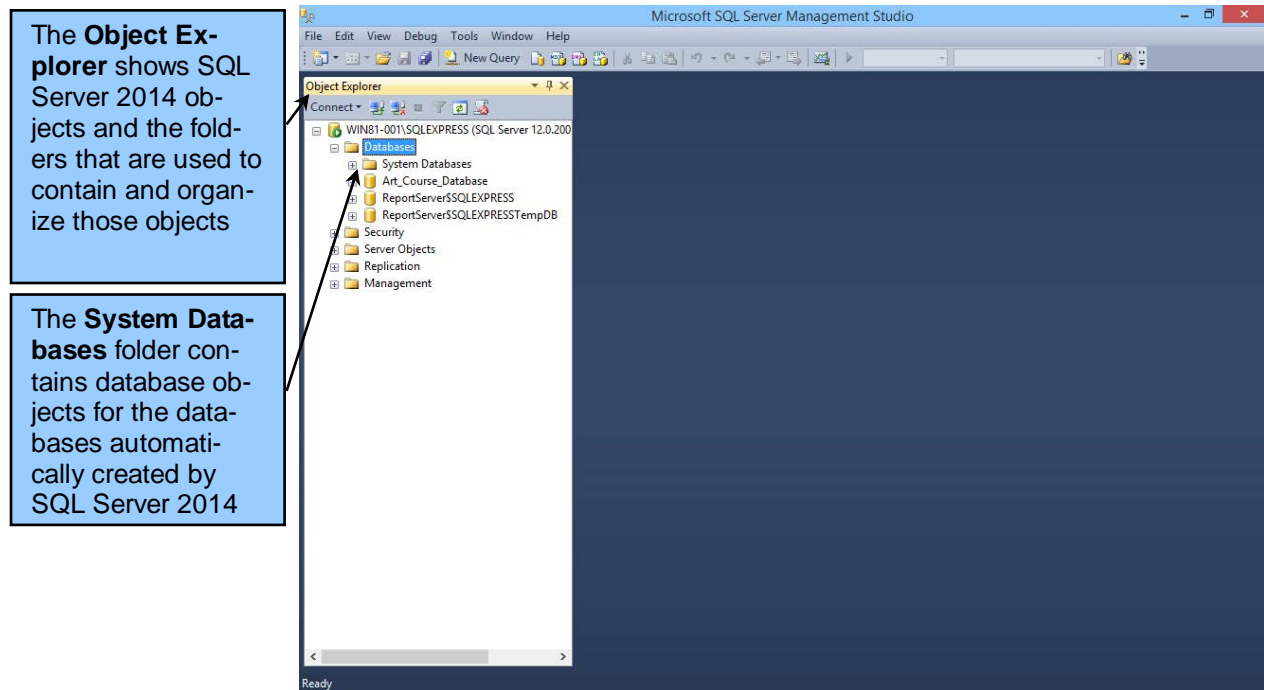
Figure A-2 — The Microsoft SQL Server Management Studio in Windows 7

In the Microsoft SQL Server Management Studio, you should have an **Object Explorer** window on the left side. The Object Explorer displays SQL Server 2014 objects (such as databases and tables) and the folders (with labels such as Databases and Security) that are used to organize the presentation of the objects. The folders are considered to be objects, and, therefore, the window contains an expandable set of objects. In Figure A-2, the **Databases folder** has been expanded to show that it contains a folder named System Databases and a database named Art-Course-Database. Also note the databases named **ReportServerSQLEXPRESS** and **ReportServerSQLEXPRESSTempDB**—these are databases used by the **SQL Server 2014 Report Services** engine that is provided as part of the SQL Server 2014 Express Advanced package.

Creating a Database in SQL Server

To create an SQL Server database, right-click the **Databases** object in the Object Explorer to display the shortcut menu and then click **New Database**, as shown in Figure A-3.

The **New Database** dialog box appears, as shown in Figure A-4. Type the name of the new database—in this example, **WPC** (for Wedgewood Pacific Corporation)—in the Database Name text box and then click the **OK** button to create the new database. After you create the database, the new database name appears in the Databases folder object. Figure A-5 shows the WPC database folder object selected and expanded to show the objects within the database.

Right-click **Data-bases** to display the shortcut menu

Click **New Data-base** to create a new database
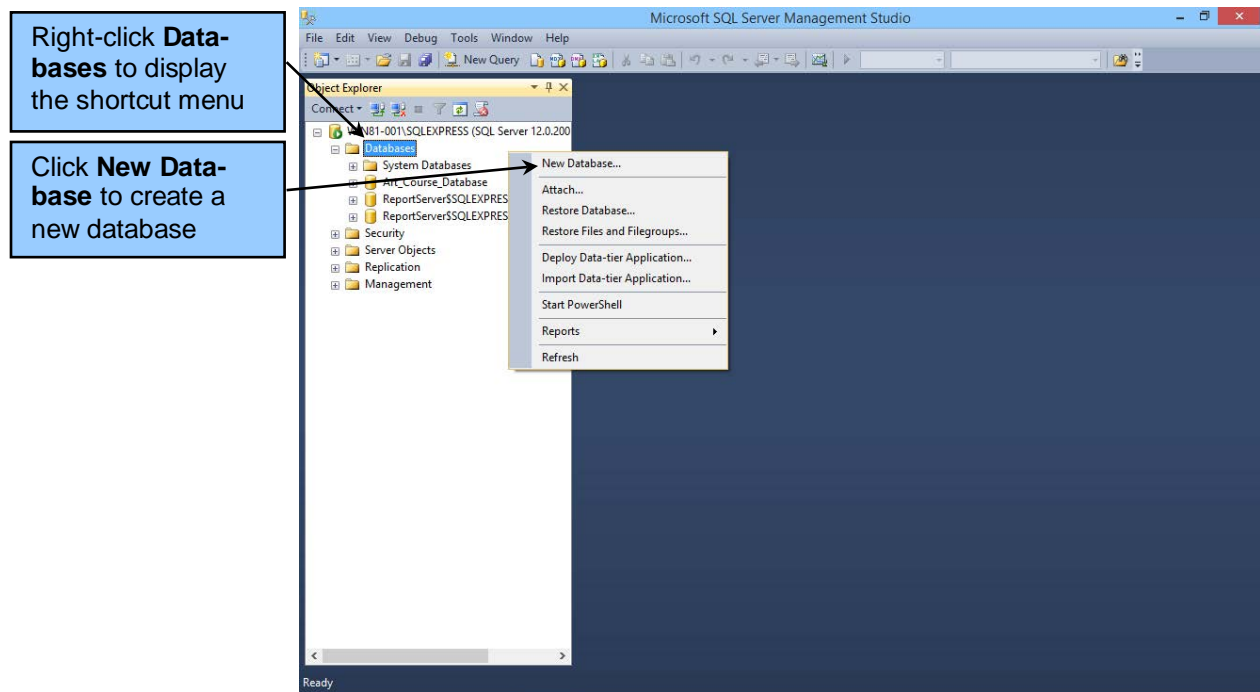
Figure A-3 — The New Database Command

The **New Data-base** dialog box

Type the new data-base name here

Click the **OK** button to create the new database
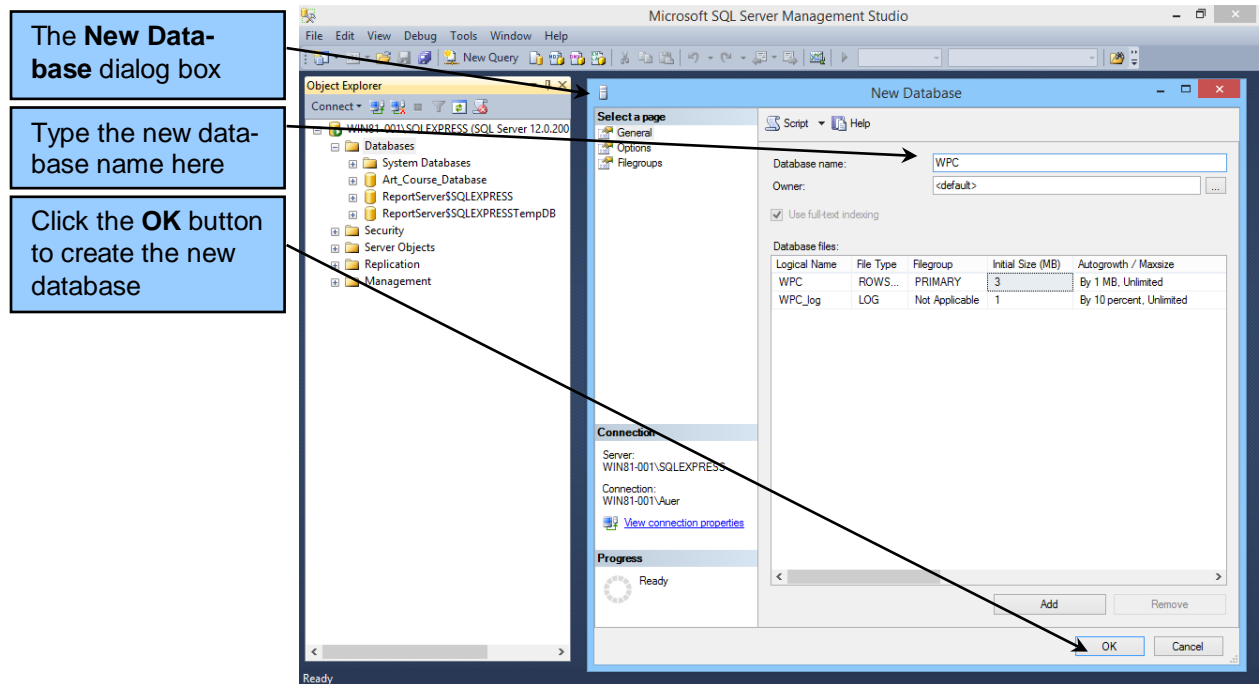
Figure A-4 — Naming the New Database

The **WPC data-base** in the Object Explorer, expanded to show the folder structure that will be used to contain the database objects
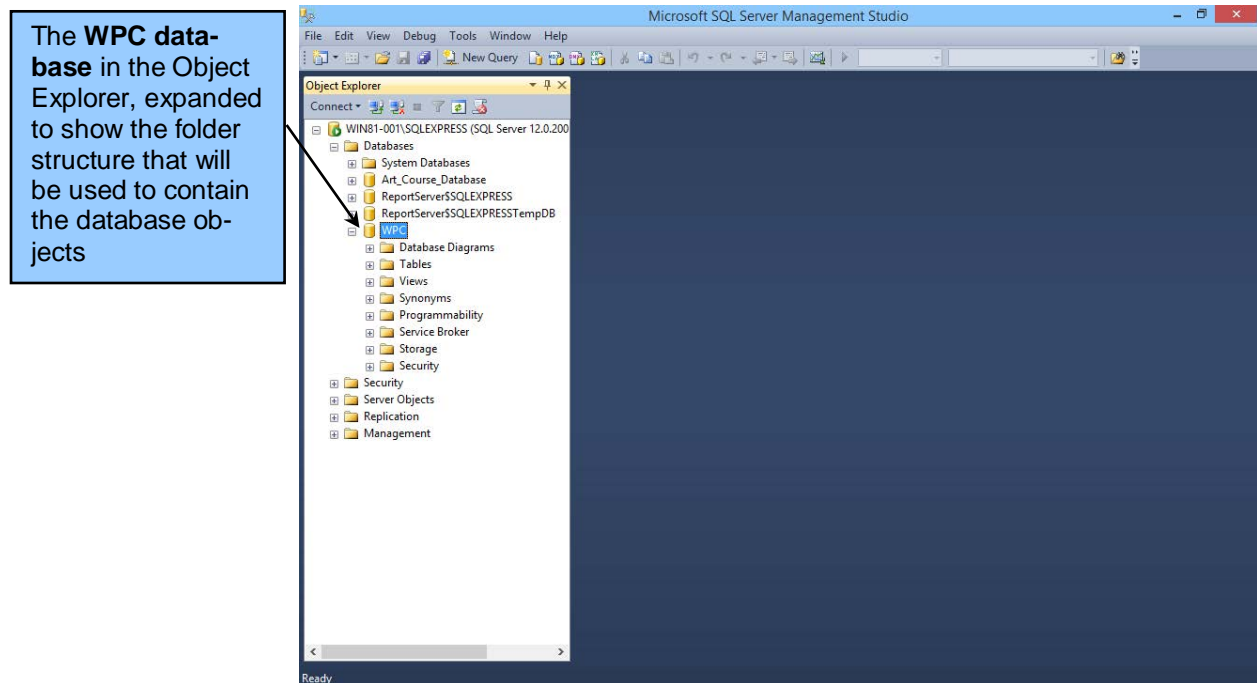
Figure A-5 — The New Database in the Object Explorer

Working with SQL Statements in SQL Server

In SQL Server, SQL statements can be run individually or as part of a related group of SQL statements known as an **SQL script**. SQL scripts are efficient for processing groups of SQL statements. For example, you could create a set of CREATE TABLE commands to build a new database structure as a script, or you could create a set of INSERT commands to use when data needs to be added to a table as a script.

Working with SQL Scripts in SQL Server

You can create scripts in the SQL Server Management Studio or in any ASCII text editor. In the Windows operating system, the Notepad text editor is a good choice if you're not using SQL Server Management Studio itself. Regardless of which text editor you use, save your scripts with the file extension ***.sql** so that SQL Server recognizes them. By default, SQL Server Management Studio looks for scripts in a special folder that is created in the user's My Documents folder when the program is installed on a personal workstation. The folder structure created by the installation process is shown in Figure A-6, along with a *newly created* folder named **Projects** that we have created. SQL scripts should be stored in the Projects folder, in a subfolder named for the database.

By The Way

Using the Microsoft SQL Server Management Studio for Microsoft SQL Server 2014, we have to create the Projects folder. In some previous versions of the Microsoft SQL Server Management Studio, the Projects folder is automatically created (along with the SQL Server Management Studio folder and the other subfolders).

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

SQL Commands to Create Table Structures

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

To create a new document for our SQL script, click the New Query button shown in Figure A-8. A new tabbed document window, in this case named SQLQuery1.sql, is displayed to the right of the Object Browser. The **SQL Editor toolbar** is also displayed, which contains tools useful when editing text. One of
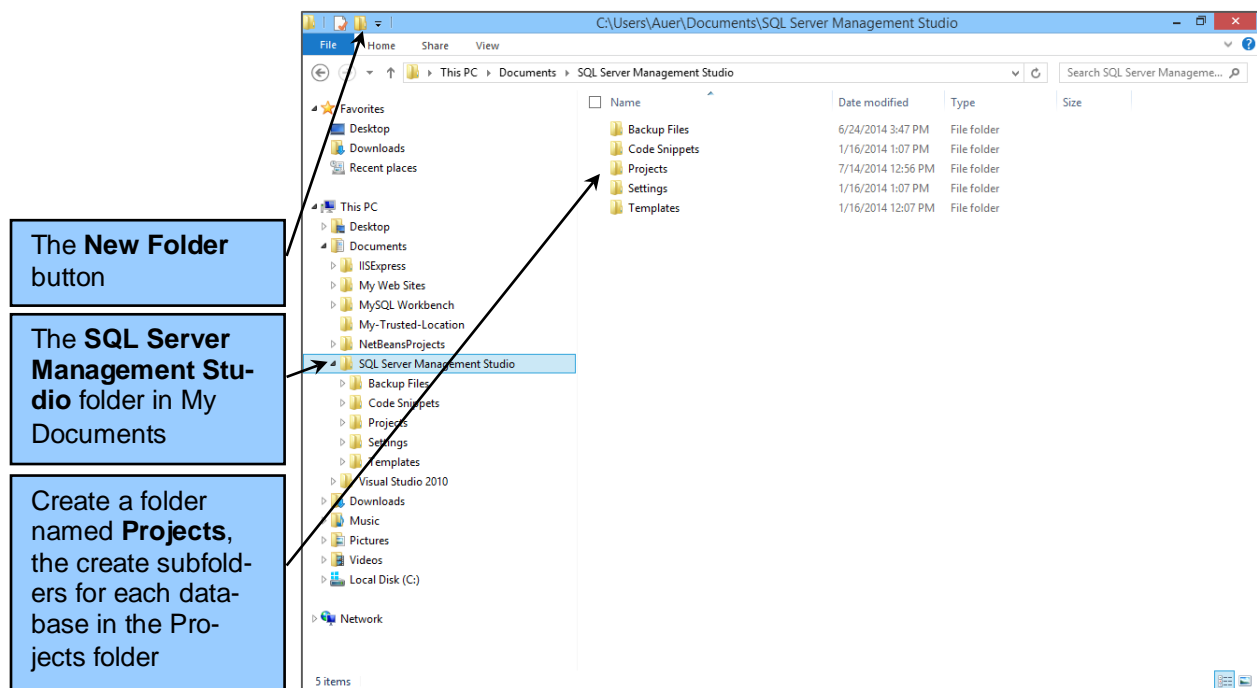
Figure A-6 — The Microsoft SQL Server Management Studio Folder in Windows Explorer

```sql
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 IDENTITY (1, 1),
    FirstName         Char(25)       NOT NULL,
    LastName          Char(25)       NOT NULL,
    Department        Char(35)       NOT NULL DEFAULT 'Human Resources',
    Phone             Char(12)       NULL,
    Email             VarChar(100)   NOT NULL UNIQUE,
    CONSTRAINT        EMPLOYEE_PK    PRIMARY KEY(EmployeeNumber),
    CONSTRAINT        EMP_DEPART_FK  FOREIGN KEY(Department)
                      REFERENCES DEPARTMENT(DepartmentName)
                          ON UPDATE CASCADE
    );

CREATE   TABLE PROJECT (
    ProjectID         Int            NOT NULL IDENTITY (1000, 100),
    ProjectName       Char(50)       NOT NULL,
    Department        Char(35)       NOT NULL,
    MaxHours          Numeric(8,2)   NOT NULL DEFAULT 100,
    StartDate         Date           NULL,
    EndDate           Date           NULL,
    CONSTRAINT        PROJECT_PK     PRIMARY KEY (ProjectID),
    CONSTRAINT        PROJ_DEPART_FK FOREIGN KEY(Department)
                      REFERENCES DEPARTMENT(DepartmentName)
                          ON UPDATE CASCADE
    );

CREATE   TABLE ASSIGNMENT (
    ProjectID         Int            NOT NULL,
    EmployeeNumber    Int            NOT NULL,
    HoursWorked       Numeric(6,2)   NULL,
    CONSTRAINT        ASSIGNMENT_PK  PRIMARY KEY (ProjectID, EmployeeNumber),
    CONSTRAINT        ASSIGN_PROJ_FK FOREIGN KEY (ProjectID)
                      REFERENCES PROJECT (ProjectID)
                          ON UPDATE NO ACTION
                          ON DELETE CASCADE,
    CONSTRAINT        ASSIGN_EMP_FK  FOREIGN KEY (EmployeeNumber)
                      REFERENCES EMPLOYEE (EmployeeNumber)
                          ON UPDATE NO ACTION
                          ON DELETE NO ACTION
    );
```

Figure A-7 — The Figure 3-7 SQL Statements to Create the WPC Database Tables
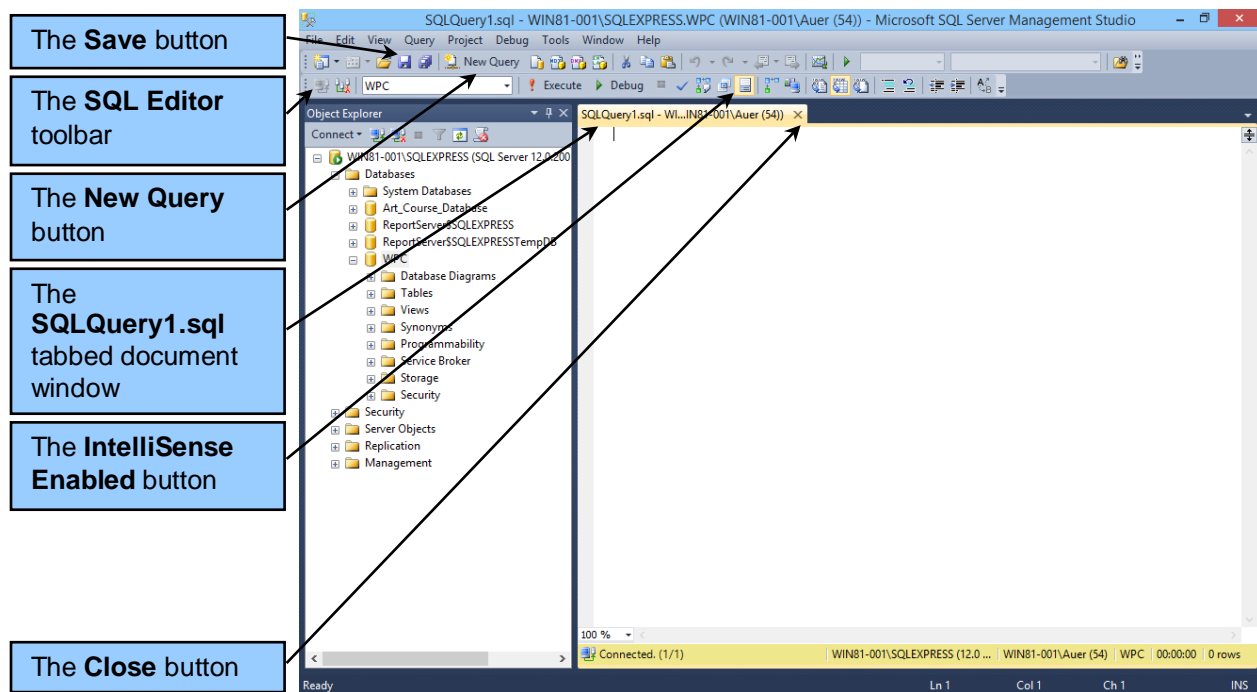
Figure A-8 — The SQLQuery1.sql Tabbed Document Window

these is the **IntelliSense Enabled button**, which may be useful to you as it provides some outlining and auto-completion capabilities. However, you may also find it more confusing to see the notations and auto-completion suggestions that are displayed (I know I often do!), and you can simply click the IntelliSense button to disable this feature. Also note the location of the **Close button** used to close the tabbed document window.

Figure A-9 shows:

- The text in Figure A-7 typed into the tabbed document window together with a header of comments to document the file content—note the color coding.
- This text saved with the name **DBC-e07-MSSQL-WPC-Create-Tables.sql** (you can use a shorter name if you prefer, such as just *WPC-Create-Tables.sql*). This was done by clicking the **Save** button shown in Figure A-8, browsing to the Projects folder, creating a subfolder named WPC Scripts, and then naming and saving the file.
- The **Available Databases** drop-down list, which is used to select the **active database**—the one that the script will be run against.
- The **Parse** button, which is used to test the SQL code before actually running it.
- The **Execute** button, which is used to actually run the SQL code in the document window.

Test your code until the parser doesn't find any errors, save the script again, and then execute the script. Figure A-10 shows the tables in the WPC database.

The **Available Databases** drop-down list—select the **active database** from this list

The **Execute** button

The **Parse** button

The saved **DBC-e07-MSSQL-WPC-Create-Tables.sql** code—note the comments added as a header
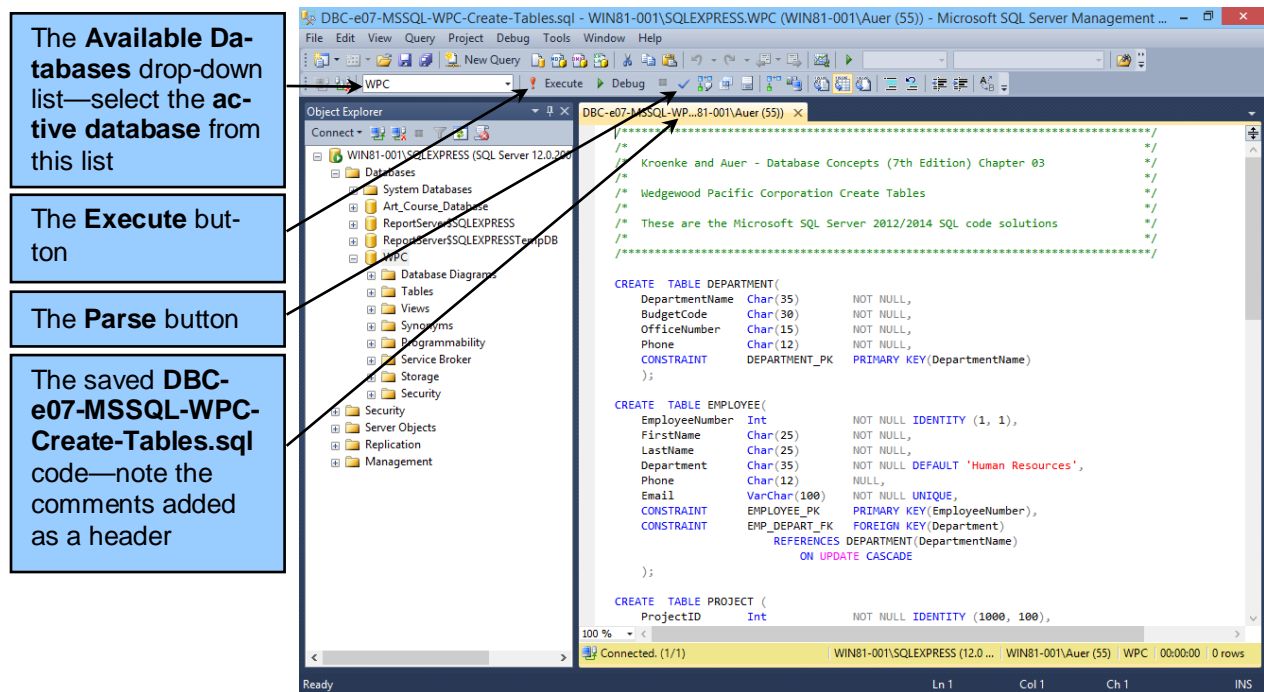
Figure A-9 — The SQL Script to Create the WPC Database Tables

The objects representing the tables created by the SQL script in the expanded Tables folder—*dbo* stands for *database owner*

Output messages are shown here—either messages that the commands were successful or appropriate error messages

Figure A-10 — The SQL Script Results—The WPC Tables

By The Way

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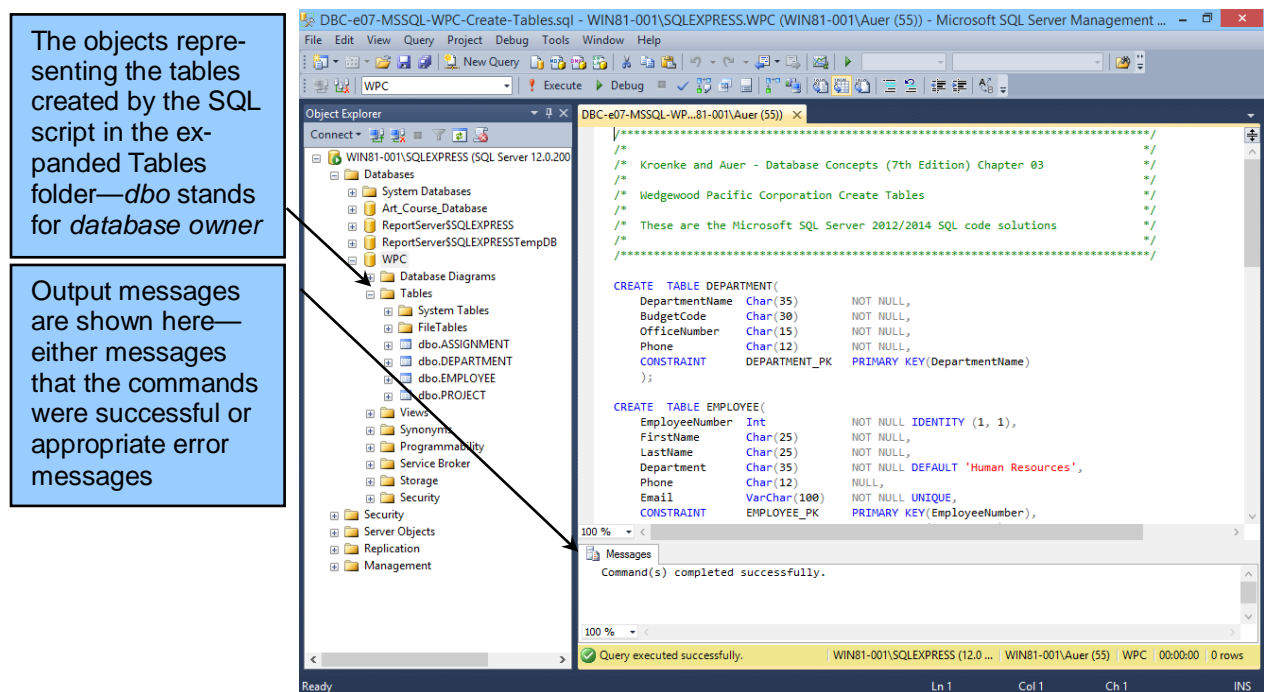 such words whenever possible—*ProjectName*, *StartDate*, *ItemTransaction*—or enclose the unmodified words in square brackets—*[Name], [Date], [Transaction]*—if you must use them.

SQL Commands to Insert Database Data

The SQL statements needed to insert data into the tables you have created are the same as the ones shown in Figure 3-12, which are repeated in Figure A-11 on the next page. You also need to put this data set into a script file, save it, *be sure the WPC database is selected as the active database*, and then run it as an SQL script.

Working with SQL Queries in SQL Server

Now that we've created and populated the WPC database, we can run SQL queries against the data. While scripts are good for large sets of SQL commands that need to be run together, most SQL queries are run as single commands. To run a query, be sure the Databases folder is expanded in the Object Browser so that you can see the database names. Click the **New Query** button in the Standard toolbar, as shown in Figure A-8, to display a new tabbed SQL window along with the SQL Editor toolbar. Then specify the database you want to query by selecting the database name in the Available Databases drop-down list to select it.

By now you should be familiar with using the SQL query tabbed document window and running SQL scripts. So, to create and run the SQL query, use the new SQL query window to:

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

The query results appear in a tabbed Results window below the query window in a spreadsheet-style display, as shown in Figure A-12. You can adjust the size of the query window and the Results window, and you can modify the column widths in the results display by using standard Windows drag-and-drop techniques to help make more data visible. You can also have multiple queries open at the same time—clicking the New Query button again opens another tabbed query window.

```
/*****    DEPARTMENT DATA    ***************************************************/

INSERT INTO DEPARTMENT VALUES(
    'Administration', 'BC-100-10', 'BLDG01-300', '360-285-8100');
INSERT INTO DEPARTMENT VALUES('Legal', 'BC-200-10', 'BLDG01-200', '360-285-8200');
INSERT INTO DEPARTMENT VALUES('Accounting', 'BC-300-10', 'BLDG01-100', '360-285-8300');
INSERT INTO DEPARTMENT VALUES('Finance', 'BC-400-10', 'BLDG01-140', '360-285-8400');
INSERT INTO DEPARTMENT VALUES(
    'Human Resources', 'BC-500-10', 'BLDG01-180', '360-285-8500');
INSERT INTO DEPARTMENT VALUES('Production', 'BC-600-10', 'BLDG02-100', '360-287-8600');
INSERT INTO DEPARTMENT VALUES('Marketing', 'BC-700-10', 'BLDG02-200', '360-287-8700');
INSERT INTO DEPARTMENT VALUES('InfoSystems', 'BC-800-10', 'BLDG02-270', '360-287-8800');


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

INSERT INTO EMPLOYEE VALUES(
    'Mary', 'Jacobs', 'Administration', '360-285-8110', 'Mary.Jacobs@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Rosalie', 'Jackson', 'Administration', '360-285-8120', 'Rosalie.Jackson@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Richard', 'Bandalone', 'Legal', '360-285-8210', 'Richard.Bandalone@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Tom', 'Caruthers', 'Accounting', '360-285-8310', 'Tom.Caruthers@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Heather', 'Jones', 'Accounting', '360-285-8320', 'Heather.Jones@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Mary', 'Abernathy', 'Finance', '360-285-8410', 'Mary.Abernathy@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'George', 'Smith', 'Human Resources', '360-285-8510', 'George.Smith@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Tom', 'Jackson', 'Production', '360-287-8610', 'Tom.Jackson@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'George', 'Jones', 'Production', '360-287-8620', 'George.Jones@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Ken', 'Numoto', 'Marketing', '360-287-8710', 'Ken.Numoto@WPC.com');
INSERT INTO EMPLOYEE(FirstName, LastName, Department, Email)
    VALUES('James', 'Nestor', 'InfoSystems', 'James.Nestor@WPC.com');
INSERT INTO EMPLOYEE VALUES(
    'Rick', 'Brown', 'InfoSystems', '360-287-8820', 'Rick.Brown@WPC.com');

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


INSERT INTO PROJECT VALUES(
    '2014 Q3 Product Plan', 'Marketing', 135.00, '10-MAY-14', '15-JUN-14');
INSERT INTO PROJECT VALUES(
    '2014 Q3 Portfolio Analysis', 'Finance', 120.00, '05-JUL-14', '25-JUL-14');
INSERT INTO PROJECT VALUES(
    '2014 Q3 Tax Preparation', 'Accounting', 145.00, '10-AUG-14', '15-OCT-14');
```

Figure A-11 — The Figure 3-12 SQL Statements to Populate the WPC Database Tables

```sql
INSERT INTO PROJECT VALUES(
   '2014 Q4 Product Plan', 'Marketing', 150.00, '10-AUG-14', '15-SEP-14');
INSERT INTO PROJECT (ProjectName, Department, MaxHours, StartDate)
   VALUES('2014 Q4 Portfolio Analysis', 'Finance', 140.00, '05-OCT-14');

/*****    ASSIGNMENT DATA    **************************************************/

INSERT INTO ASSIGNMENT VALUES(1000, 1, 30.0);
INSERT INTO ASSIGNMENT VALUES(1000, 8, 75.0);
INSERT INTO ASSIGNMENT VALUES(1000, 10, 55.0);
INSERT INTO ASSIGNMENT VALUES(1100, 4, 40.0);
INSERT INTO ASSIGNMENT VALUES(1100, 6, 45.0);
INSERT INTO ASSIGNMENT VALUES(1200, 1, 25.0);
INSERT INTO ASSIGNMENT VALUES(1200, 2, 20.0);
INSERT INTO ASSIGNMENT VALUES(1200, 4, 45.0);
INSERT INTO ASSIGNMENT VALUES(1200, 5, 40.0);
INSERT INTO ASSIGNMENT VALUES(1300, 1, 35.0);
INSERT INTO ASSIGNMENT VALUES(1300, 8, 80.0);
INSERT INTO ASSIGNMENT VALUES(1300, 10, 50.0);
INSERT INTO ASSIGNMENT VALUES(1400, 4, 15.0);
INSERT INTO ASSIGNMENT VALUES(1400, 5, 10.0);
INSERT INTO ASSIGNMENT VALUES(1400, 6, 27.5);


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

Figure A-11 — The Figure 3-12 SQL Statements to Populate the WPC Database Tables (Continued)

The SQL query—the asterisk (*) on the window tab means that the text has been changed, but not saved, and the asterisk (*) in the SQL query means that all columns will be returned in the query results

The tabbed Results window—you can adjust window boundaries and column widths by using standard Windows drag-and-drop techniques
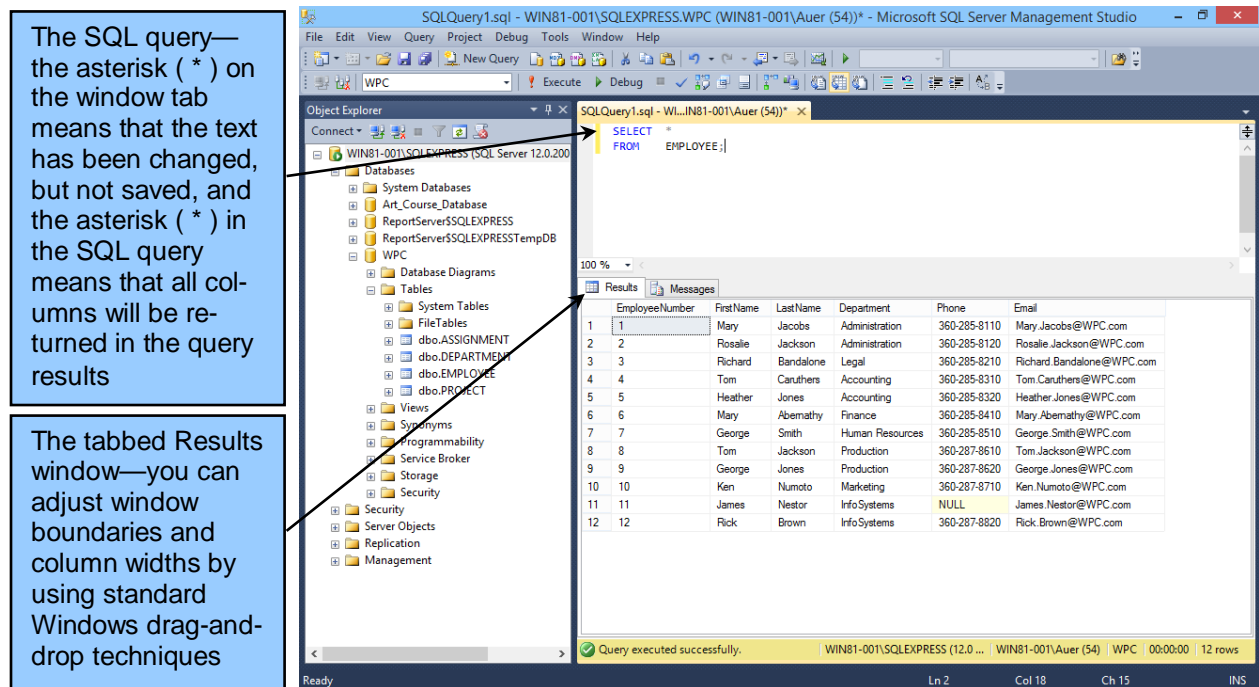
Figure A-12 — The SQL Query Results

How do I Install the Microsoft SQL Server 2014 ODBC Client?

For SQL Server 2014, you do not have to take any extra steps to install OBDC support. The ODBC Driver 11 for SQL Server is automatically installed and available for use, as shown in Figure 7-9.

Additional Documentation for SQL Server Express Edition

To get access to SQL Server documentation, use **SQL Server 2014 Books Online** at http://msdn.microsoft.com/en-us/library/ms130214.aspx.

Key Terms

*.sql	**Active database**
Available Databases drop-down list	**Close button**
Connect button	**Connect to Server dialog box**
Databases folder object	**Execute button**
IntelliSense Enabled button	**Messages window**
Microsoft SQL Server	**Microsoft .NET Framework 3.5 Service Pack 1**
Microsoft Windows Installer 4.5	**New Database dialog box**
New Query button	**Object Explorer window**
Parse button	**Results window**
ReportServerSQLEXPRESS	**ReportServerSQLEXPRESSTempDB**
SQL Editor toolbar	**SQL Server 2005 Express Edition**
SQL Server 2014 Express	**SQL Server 2008/2008 R2 Express Advanced**
SQL Server 2014 Express Advanced	**SQL Server 2014 Report Services**
SQL Server 2014 Management Studio	**SQL script**
Standard toolbar	**Windows Authentication**

Review Questions

A.1   What is SQL Server 2014 Express?

A.2   What is the primary advantage of using SQL Server 2014 Express instead of Microsoft Access?

A.3   What is the set of Microsoft programs that are recommended as a necessary set of SQL Server 2014 Express Advanced software products? In what order should you install these products?

A.4   What is the purpose of Microsoft SQL Server Management Studio?

A.5   How do you create a new database in SQL Server 2014?

A.6   How do you specify the active database in SQL Server 2014?

A.7   What is an SQL script? What types of SQL statements and commands can you run more efficiently as scripts?

A.8   What tool(s) can be used to create a script?

A.9   What file extension should you use for SQL scripts?

A.10   How do you open and run a script in SQL Server?

A.11   How do you create and run an SQL query in SQL Server?

Exercises

A.12   If you haven't already done so, download and install SQL Server 2014 Express Advanced as described in the text. Use the default settings for the installation. Be sure that the Microsoft SQL Server Management Studio is correctly installed.

A.13   If you haven't already done so, work through the steps described in this appendix to create and populate the WPC database.

A.14   Using SQL Server 2014 and the Microsoft SQL Server Management Studio, run the following SQL queries in the "SQL for Relational Queries" section of Chapter 3:

         SQL-QUERY-CH03-01 through SQL-QUERY-CH03-31

         SQL-QUERY-CH03-34 through SQL-QUERY-CH03-49

Save each query as follows:

- • Create and run each query in SQL Server Management Studio.

- • After you have run each query, use the **File | Save SQLQuery#.sql As** command to save the query (The # sign in the name changes as you create different queries.) By default, SQL Server saves each file as an SQL file with the file extension *.sql. Use this default setting unless your instructor tells you to use a different extension. Name your queries in *numerical sequence*, starting with the file name MSSQL-SQL-Query-Ch03-01.sql.

A.15   Use Microsoft SQL Server Management Studio to run one or more of the saved SQL queries you created in question A.14:

- • Open a query using the **Open File** button (or by selecting the **File | Open File** menu command). Note that the query is opened in a tabbed query window. Run the query.

- • Use the **Open File** button (or the **File | Open File** menu command) to open and run another query in another tabbed window.

- • Experiment with opening and closing windows and running various queries in these windows.

A.16   Complete exercise 3.63 using SQL Server 2014 and the Microsoft SQL Server Management Studio. Start each saved query name with MSSQL- and use the default .sql file extension. (The first saved query name should be MSSQL-SQL-Query-AWE-3-1-A.sql.)

A.17   Complete exercise 3.64 using SQL Server 2014 and the Microsoft SQL Server Management Studio. Start the saved query name with MSSQL- and use the default .sql file extension. The saved query name will be MSSQL-SQL-Query-AWE-3-3-E.sql.