

Database Processing

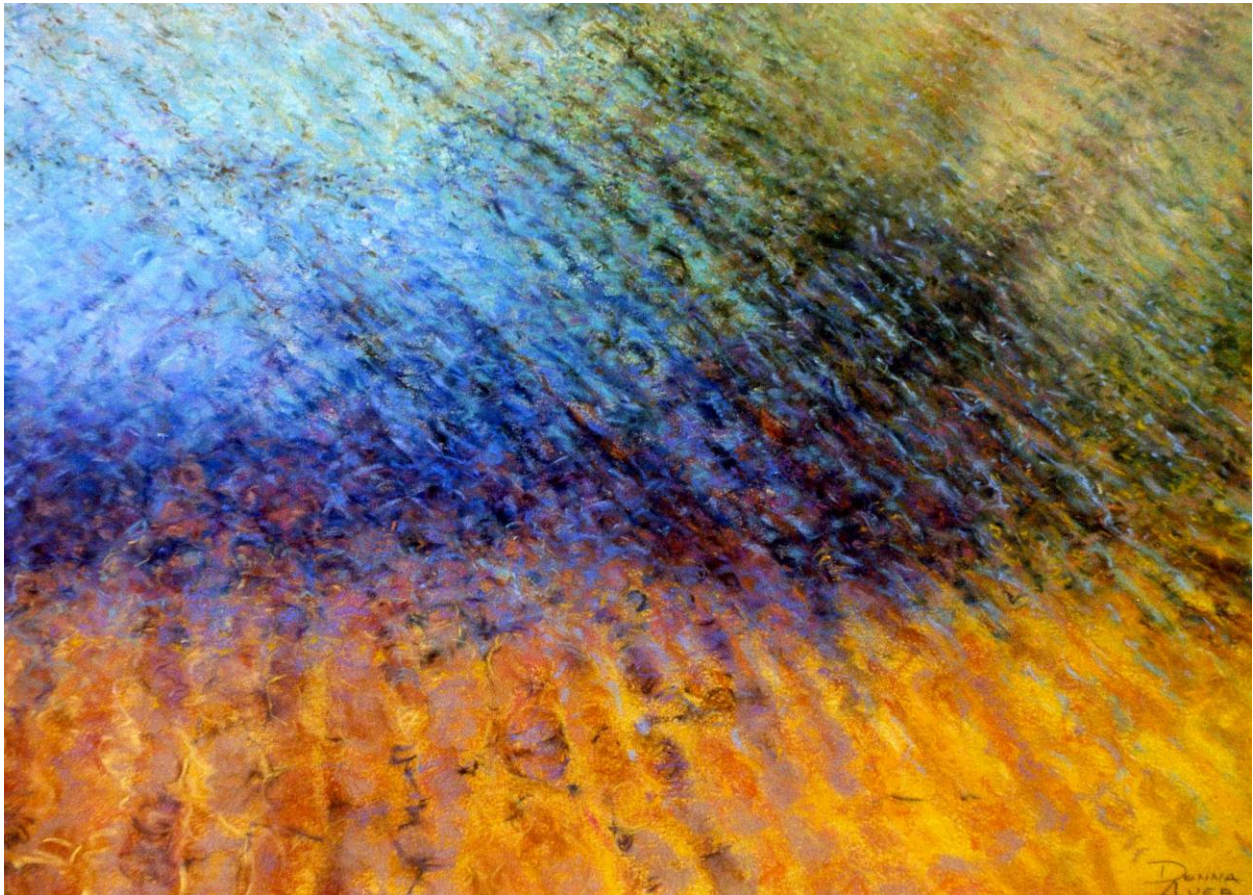
Fundamentals, Design, and Implementation

14th Edition

David M. Kroenke • David J. Auer

Online Appendix A

Getting Started with Microsoft Access 2013



Vice President, Business Publishing: Donna Battista
Editor in Chief: Stephanie Wall
Acquisitions Editor: Nicole Sam
Program Manager Team Lead: Ashley Santora
Program Manager: Denise Weiss
Editorial Assistant: Olivia Vignone
Vice President, Product Marketing: Maggie Moylan
Director of Marketing, Digital Services and Products:
 Jeanette Koskinas
Executive Product Marketing Manager: Anne Fahlgren
Field Marketing Manager: Lenny Ann Raper
Senior Strategic Marketing Manager: Erin Gardner
Product Marketing Assistant: Jessica Quazza
Project Manager Team Lead: Jeff Holcomb
Project Manager: Ilene Kahn
Operations Specialist: Diane Peirano
Senior Art Director: Janet Slowik

Text Designer: Integra Software Services Pvt. Ltd.
Cover Designer: Integra Software Services Pvt. Ltd.
Cover Art: Donna Auer
Vice President, Director of Digital Strategy & Assessment: Paul Gentile
Manager of Learning Applications: Paul Deluca
Digital Editor: Brian Surette
Digital Studio Manager: Diane Lombardo
Digital Studio Project Manager: Robin Lazrus
Digital Studio Project Manager: Alana Coles
Digital Studio Project Manager: Monique Lawrence
Digital Studio Project Manager: Regina DaSilva
Full-Service Project Management and Composition: Integra Software Services Pvt. Ltd.
Printer/Binder: RRD Willard
Cover Printer: Phoenix Color/Hagerstown
Text Font: 10/12 Mentor Std Light

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on the appropriate page within text.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® Windows®, and Microsoft Office® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

MySQL®, the MySQL Command Line Client®, the MySQL Workbench®, and the MySQL Connector/ODBC® are registered trademarks of Sun Microsystems, Inc./Oracle Corporation. Screenshots and icons reprinted with permission of Oracle Corporation. This book is not sponsored or endorsed by or affiliated with Oracle Corporation.

Oracle Database 12c and Oracle Database Express Edition 11g Release 2 2014 by Oracle Corporation. Reprinted with permission. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Mozilla 35.104 and Mozilla are registered trademarks of the Mozilla Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

PHP is copyright The PHP Group 1999–2012, and is used under the terms of the PHP Public License v3.01 available at http://www.php.net/license/3_01.txt. This book is not sponsored or endorsed by or affiliated with The PHP Group.

Copyright © 2016, 2014, 2012 by Pearson Education, Inc., 221 River Street, Hoboken, New Jersey 07030. All rights reserved. Manufactured in the United States of America. This publication is protected by Copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, 221 River Street, Hoboken, New Jersey 07030.

Many of the designations by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Library of Congress Cataloging-in-Publication Data

Kroenke, David M.
 Database processing: fundamentals, design, and implementation/David M. Kroenke, David J. Auer.—Fourteenth edition.
 pages cm
 Includes bibliographical references and index.
 ISBN 978-0-13-387670-3 (student edition)—ISBN 978-0-13-387676-5 (instructor's review copy)
 I. Database management. I. Auer, David J. II. Title.
 QA76.9.D3K76 2016
 005.74—dc23

2015005632

10 9 8 7 6 5 4 3 2 1

PEARSON

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

Appendix A – 10 9 8 7 6 5 4 3 2 1

Chapter Objectives

- To create databases in Microsoft Access 2013
- To create tables in Microsoft Access 2013
- To understand Microsoft Access 2013 data types
- To insert data into tables in Microsoft Access 2013
- To create relationships between tables in Microsoft Access 2013
- To create Query-by-Example (QBE) queries in Microsoft Access 2013
- To understand the use of the Form Wizard in Microsoft Access 2013
- To understand the use of the Report Wizard in Microsoft Access 2013

What Is the Purpose of This Appendix?

Microsoft Access 2013 is a component of Microsoft Office 2013, and is widely used as a personal database as defined in Chapter 1.¹ In this appendix, we are primarily interested in Microsoft Access's ability to create databases that we can use to test various database processing concepts.

Why Should I Learn to Use Microsoft Access 2013?

For the purposes of this book, the most important reason to learn to use Microsoft Access 2013 is that it provides a convenient method of creating databases on computers that do not have an enterprise-class DBMS such as Microsoft SQL Server 2014 Oracle Database, or MySQL 5.6 installed on them. These databases will prove particularly valuable for use with Chapter 1 and Chapter 2 of this book.

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

What Will This Appendix Teach Me?

As discussed in Chapter 1, Microsoft Access is a personal database that combines a DBMS with an application generator. The DBMS performs the standard DBMS functions of database creation, processing, and administration, whereas the application generator adds the abilities to create and store forms, reports, queries, and other application-related functions. In this appendix, we will learn to use Microsoft Access 2013 to create a new database, to create tables and relationships, to insert data into tables, and to create queries (in QBE only—SQL queries are covered in Chapter 2), forms and reports.

We will begin by creating a Microsoft Access database to store the database tables and the application forms, reports, and queries. We will build the example *Student_Class_Grade* database we used in Chapter 1 to illustrate basic relational database concepts. The tables and relationship links are shown in Figure A-1.

We can write the table structure of that database in database **schema** format as:

STUDENT (StudentNumber, LastName, FirstName, EmailAddress)

CLASS (ClassNumber, ClassName, Term, Section)

GRADE (StudentNumber, ClassNumber, Grade)

The STUDENT table

The CLASS table

The GRADE table with foreign keys—now each grade is linked back to the STUDENT and CLASS tables

StudentNumber	LastName	FirstName	EmailAddress
1	Cooke	Sam	Sam.Cooke@OurU.edu
2	Lau	Marcia	Marcia.Lau@OurU.edu
3	Harris	Lou	Lou.Harris@OurU.edu
4	Greene	Grace	Grace.Greene@OurU.edu

ClassNumber	ClassName	Term	Section
10	CHEM 101	2014-Fall	1
20	CHEM 101	2014-Fall	2
30	CHEM 101	2015-Spring	1
40	ACCT 101	2014-Fall	1
50	ACCT 101	2015-Spring	1

StudentNumber	ClassNumber	Grade
1	10	3.7
1	40	3.5
2	20	3.7
3	30	3.1
4	40	3.0
4	50	3.5
		0.0

Figure A-1 — The Student_Class_Grade Database

In the schema, table names are written in all uppercase letters (for example, STUDENT). Column names are written with the initial letter of each name capitalized (for example, Grade). If the column name is a compound name, then the initial letter of each word in the name is capitalized (for example, Student-Number).

What Is a Table Key?

Each table has a **key**, which is one or more columns that uniquely identify a row. You will learn the importance of keys, their properties, and their uses throughout this text. For now, just understand that the values of a **primary key** column identify a unique row in the table. In the schema, primary key columns in a table are underlined. Thus, the primary key of STUDENT is StudentNumber. This means that a particular value of StudentNumber, say 1, identifies one and only one row in STUDENT. Similarly, the primary key of CLASS is ClassNumber.

No single column can be a primary key for the GRADE table. A student may have several grades recorded, so StudentNumber, by itself, is not a primary key. Similarly, a class will have many students, so ClassNumber, by itself, cannot be the primary key either. However, if we assume that a student takes a class just once, then the combination (StudentNumber, ClassNumber) is a primary key.

What Are Relationships?

As described in Chapter 1, one of the characteristics of a relational database is that the rows in tables can relate to one another. For the example in Figure A-1, rows in the STUDENT table are related to rows in the GRADE table by the column StudentNumber. As shown in that figure, the student with StudentNumber 1 has earned all of the grades in the GRADE table that have a matching value of 1 in the StudentNumber column. The StudentNumber column in the CLASS table, which creates the relationship link to the STUDENT table, is called a **foreign key**. Similarly, rows in the CLASS table are related to rows in the GRADE table by the column ClassNumber, and the ClassNumber column in GRADE is also a foreign key. In the database schema, foreign key columns in a table are shown in italics.

How Do I Create a New Microsoft Access Database?

We will name our Microsoft Access database **Student_Grade_Class**. Our first step is to create a new Microsoft Access 2013 database file. Note the Microsoft Access database file format we will be using ends with the **file extension** **.accdb* (for Access Database). This file format was first introduced in Microsoft Access 2007, was maintained in Microsoft Access 2010, and remains the file format used for Microsoft Access 2013 database files.

Creating the Microsoft Access Database Student_Class_Grade

1. On the Windows 8 Start screen select the **Access 2013** app tile, as shown in Figure A-2 (If you are using Windows XP, Windows Vista, or Windows 7, then select *Start | All Programs | Microsoft Office | Microsoft Access 2013*).
 2. The Microsoft Access 2013 window appears, as shown in Figure A-3. Note that the Microsoft Access 2013 window opens with the Access *Backstage* start screen displayed, which shows (1) a list of recently opened database files, (2) the **Open Other Files** button (used to browse for other database files), and (3) a set of Microsoft Access database templates available to the user. The *Custom web app* template is selected by default, but we want to create a *blank desktop database*.
- **NOTE:** The menu commands, icon location, and file locations used here are those found when using the Microsoft Windows 8 operating system. If you are using the Microsoft Windows 7, Microsoft Windows Vista, or the Microsoft Windows XP operating system, the exact terminology will vary somewhat, but these variations will not change the required actions.
 - **NOTE:** Microsoft Access 2013 (with the Light Grey Office Theme) is used in this book, and the wording of the steps and appearance of the screenshots reflect its use. If you have a different version of Microsoft Access, there will be some differences in the step-by-step details and in what you see onscreen. However, the basic functionality is the same, and you can complete these steps using any version of Microsoft Access.

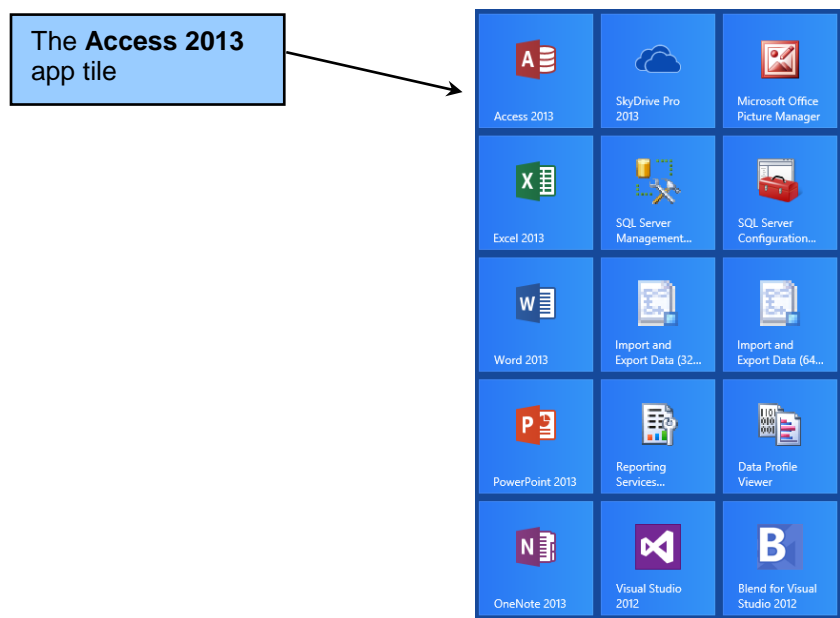


Figure A-2 — The Microsoft Access 2013 App Tile on the Windows 8 Start Screen

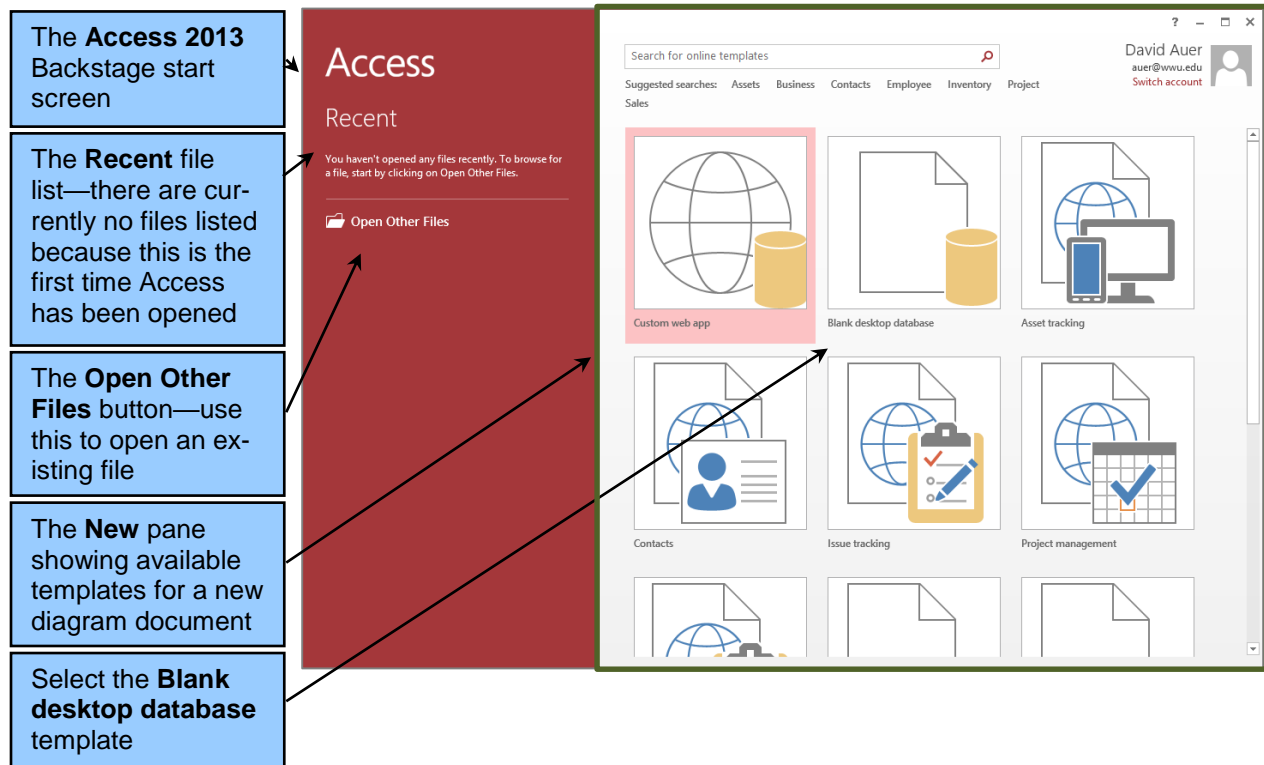


Figure A-3 — The Microsoft Access 2013 Recent File List and New Database Templates

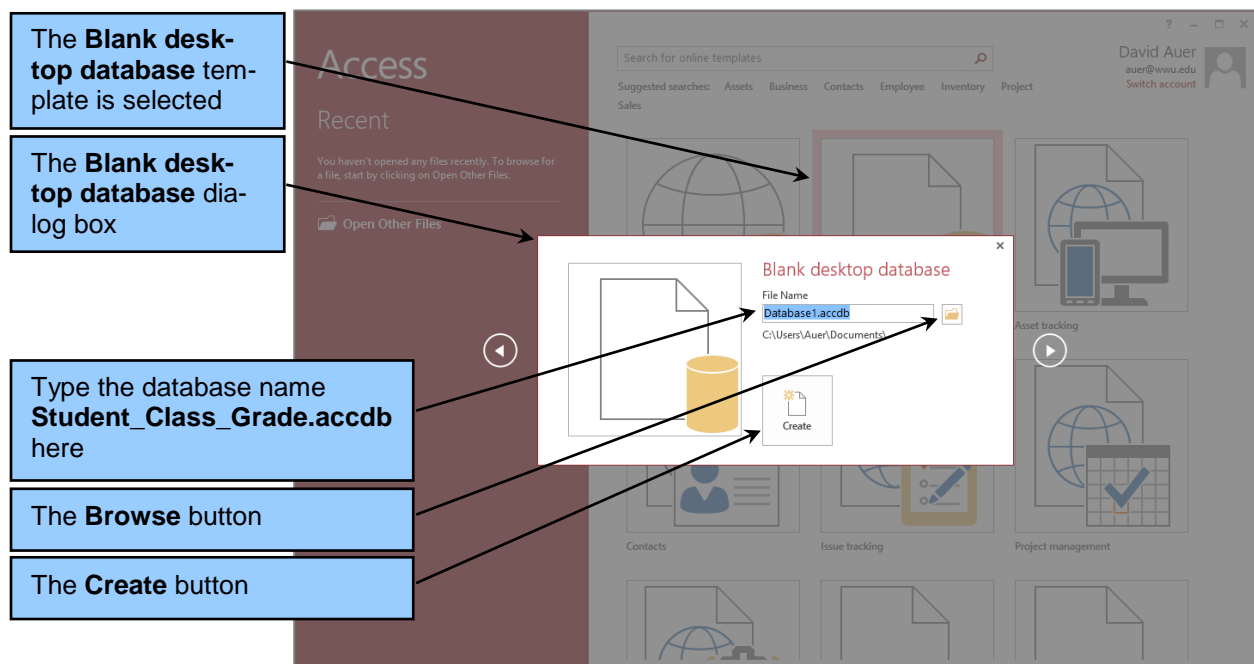


Figure A-4 — The Blank Desktop Database Dialog Box

3. Click the Blank desktop database template in the New pane. The Blank desktop database dialog box is displayed, as shown in Figure A-4.
4. Type the database name **Student_Class_Grade.accdb** into the File Name text box and then click the **Create** button.
 - **NOTE:** By default, in Windows 8 the database file will be created in your *My Documents* folder in the *Documents* library folder (the *Documents* library folder contains both a *My Documents* folder and a *Public Documents* folder). If you want to create the database file in a different location, use the **Browse** button shown in Figure A-4.
 - **NOTE:** If you use the **Browse** button to browse to a different file location, you will use the File New Database dialog box to create the new database file. Once you have browsed to the correct folder, type the database name in the File Name text box of the File New Database dialog box, and then click the OK button to create the new database.
5. The new database appears, as shown in Figure A-5. The Microsoft Access window itself is now named **Student_Grade_Class: Database** (followed by the fully qualified database file name *C:\Users\{UserName}\Documents\Student_Grade_Class.accdb* and the file format *Access 2007 - 2013 file format*, only parts of which are visible in the screen shot in Figure A-5).
 - **NOTE:** The reference to the Microsoft Access 2007-2013 file format in the window name indicates that the database is stored as an **.accdb* file, which is the Microsoft Access database file format introduced with Microsoft Access 2007, and used in Microsoft Access 2010 and now in Microsoft Access 2013. Prior versions of Microsoft Access used the **.mdb* file format (which can be opened and manipulated in Access 2013). Thus, Microsoft Access 2013 does not introduce a new file format, but continues to use the Microsoft Access 2007 **.accdb* file format.
6. Note that because this is a new database, Microsoft Access 2013 has assumed that you will want to immediately create a new table. Therefore, a new table named **Table1** is displayed in Datasheet view in the document window. We do *not* want this table open at this time, so click the **Close** document button shown in Figure A-5.
7. The Microsoft Access 2013 window with the new database appears, as shown in Figure A-6. You can see most of the features of the Microsoft Office Fluent user interface in this window.

What Is the Microsoft Office Fluent User Interface?

Microsoft Access 2013 uses the **Microsoft Office Fluent user interface** found in most (but not all) of the Microsoft Office 2013 applications. The major features of the interface can be seen in Figure A-6. Also note that the complete Microsoft Access 2013 database name of **Student_Grade_Class : Database - C:\Users\{UserName}\Documents\Student_Grade_Class.accdb (Access 2007 - 2013 file format)** is visible at the top of Figure A-6.

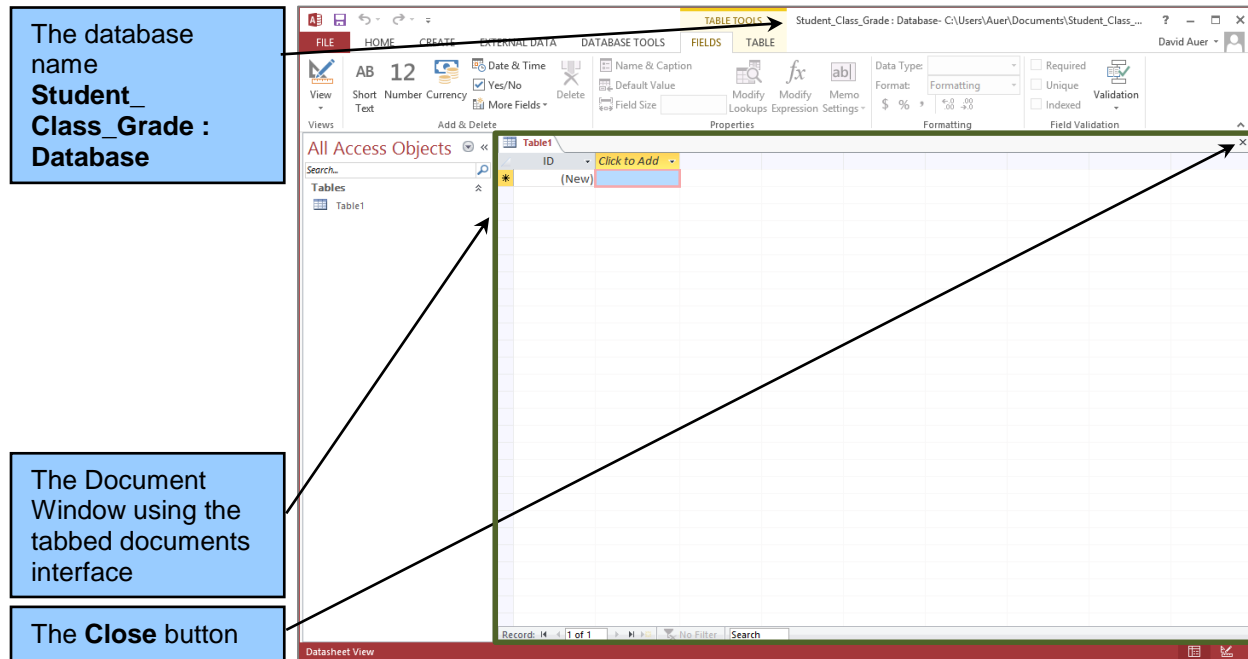


Figure A-5 — The New Microsoft Access Database

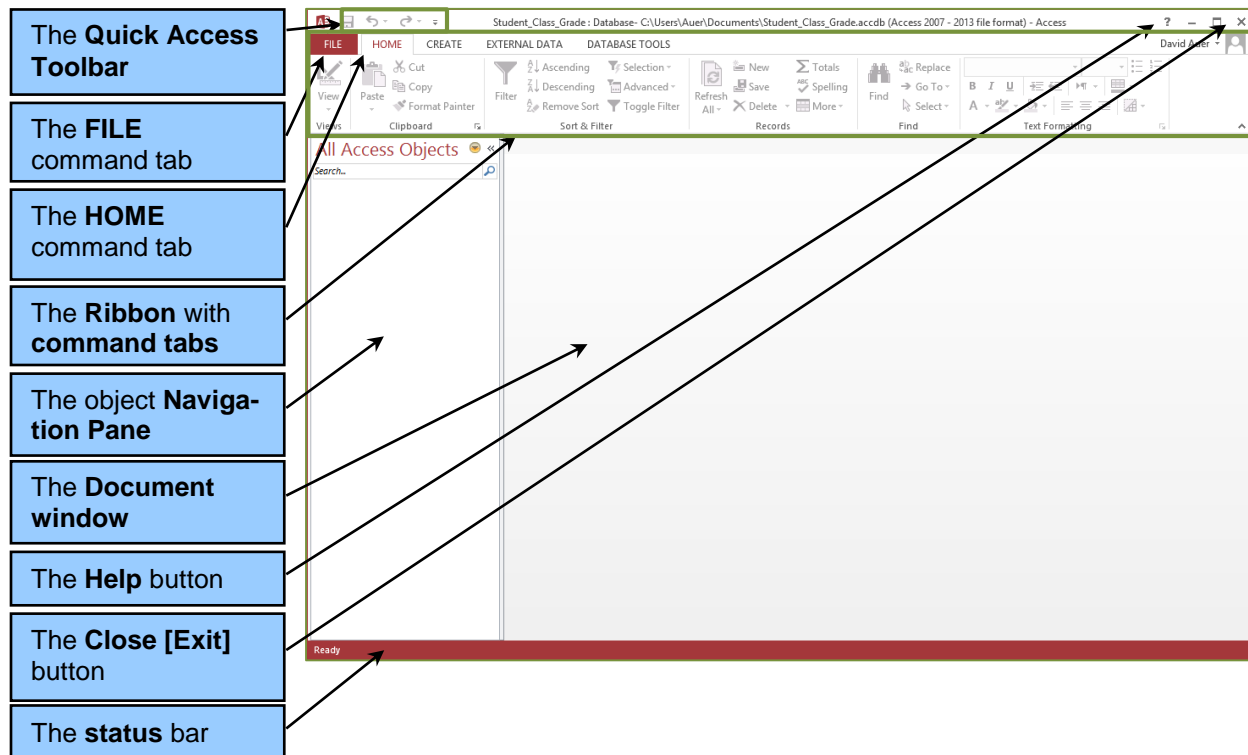


Figure A-6 — The Microsoft Office Fluent User Interface

The Ribbon and Command Tabs

The tabbed Ribbon, or just **Ribbon**, shown in Figure A-6 is the main Microsoft Access 2013 command interface. The interface provides a set of **command tabs** that you use to access tools that are grouped into sets of related commands. Each Office 2013 application has a Home tab and a set of additional tabs specific to each application.

The default Microsoft Access command tabs are the **HOME**, **CREATE**, **EXTERNAL DATA**, and **DATABASE TOOLS** tabs (the **FILE** command tab is common to all Microsoft Office products, and opens the Backstage screen for each product—note that the Backstage start screen shown in Figure A-3 is *not* available via the FILE command tab, only when you initially start Microsoft Access 2013). In each command tab, the currently available commands are shown in color, and the unavailable commands are shown in gray. You will learn about the various commands as they are needed, so for now you can just become familiar with each command tab and its command groups.

Contextual Command Tabs

In addition to the basic command tabs, some Office 2013 applications, including Microsoft Access, have additional **contextual command tabs** with associated command groups. These are displayed as needed, depending on the task. You can see an example in Figure A-5, where the **TABLE TOOLS** contextual grouping of tabs adds the **FIELDS** and **TABLE** contextual command tabs into the set of command tabs available on the Ribbon. You will learn about the other contextual tabs as you encounter them.

BY THE WAY

Microsoft documentation varies about just what constitutes the Ribbon. The Quick Access Toolbar documentation is split and often includes the toolbar in the Ribbon. In this book, we will define the Ribbon to only include the command tabs and contextual command tabs.

Modifying the Quick Access Toolbar

We'll illustrate the use of the Quick Access Toolbar shown in Figure A-6 by modifying it to include a **Quick Print** button and a **Print Preview** button.

Modifying the Access Quick Access Toolbar

1. Click the **Customize Quick Access Toolbar** drop-down button shown in Figure A-7. The Customize Quick Access Toolbar drop-down list appears as shown in Figure A-7.
 2. Click **Quick Print**. The Quick Print button is added to the Quick Access Toolbar.
 3. Click the **Customize Quick Access Toolbar** drop-down button. The Customize Quick Access Toolbar drop-down list appears.
 4. Click **Print Preview**. The Print Preview button is added to the Quick Access Toolbar.
 5. The added buttons are visible in the figures shown later in this Appendix and in the book chapters, such as Figure A-8.
-

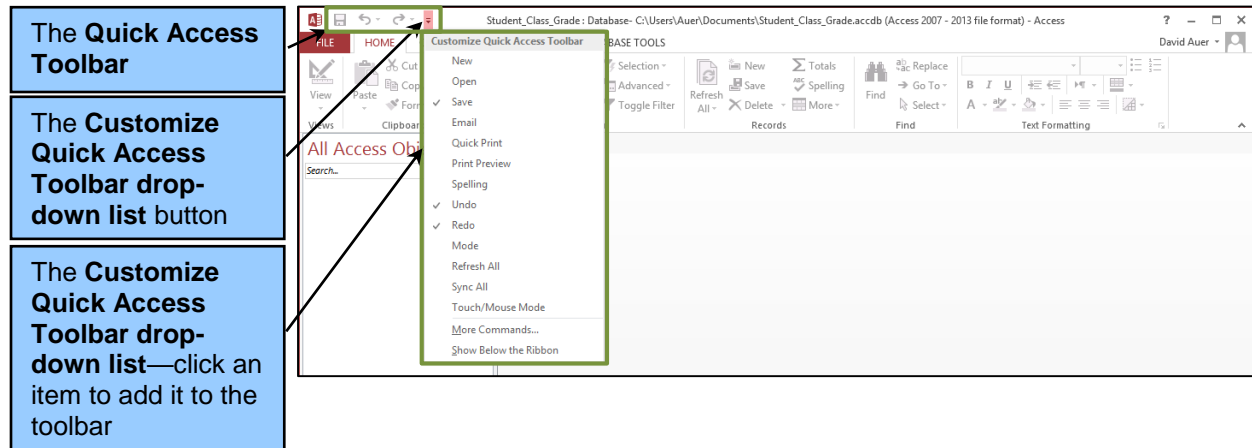


Figure A-7 — The Quick Access Toolbar

Database Objects and the Navigation Pane

Microsoft uses the term **object** as a general name for the various parts of a Microsoft Access database. Thus, a *table* is an object, a *report* is an object, a *form* is an object, and so on. Microsoft Access objects are displayed in the Microsoft Access **Navigation Pane**, as shown in Figure A-6. However, because you have not created any objects in the Student-Class-Grade database, the Navigation Pane is currently empty.

The Navigation Pane is currently labeled as *All Access Objects*, which is what we want to see displayed. We can, however, select exactly which objects will be displayed by using the **Navigation Pane drop-down list**. As shown in Figure A-8, the Navigation Pane drop-down list is controlled by the **Navigation Pane drop-down list** button. Figure A-9 shows the empty Navigation Pane, and the **Shutter Bar Open/Close** button. We can hide the Navigation Pane if we want to by clicking the Shutter Bar Open/Close button, which is displayed as a left-facing double chevron button on the upper-right corner of the Navigation Pane in Figure A-9. If we click the button, the Navigation Pane shrinks to a small band labeled *Navigation Pane* on the right side of the Microsoft Access 2013 window. The band will then display the Shutter Bar Open/Close button as a right-facing double chevron button that you can click to restore the Navigation Pane when you want to use it again.

How Do I Close a Database and Exit Microsoft Access 2013?

The **Close** button shown in Figure A-6 is actually a *close and exit button*. You can click it to close the active database and then exit the Microsoft Access program. Note that Microsoft Access actively saves most changes to a database, and it prompts you with *Save* command requests when they are needed. For example, when you close a table with modified column widths, Microsoft Access asks if you want to save the changes in the table layout. Therefore, you do not need to save Microsoft Access databases the

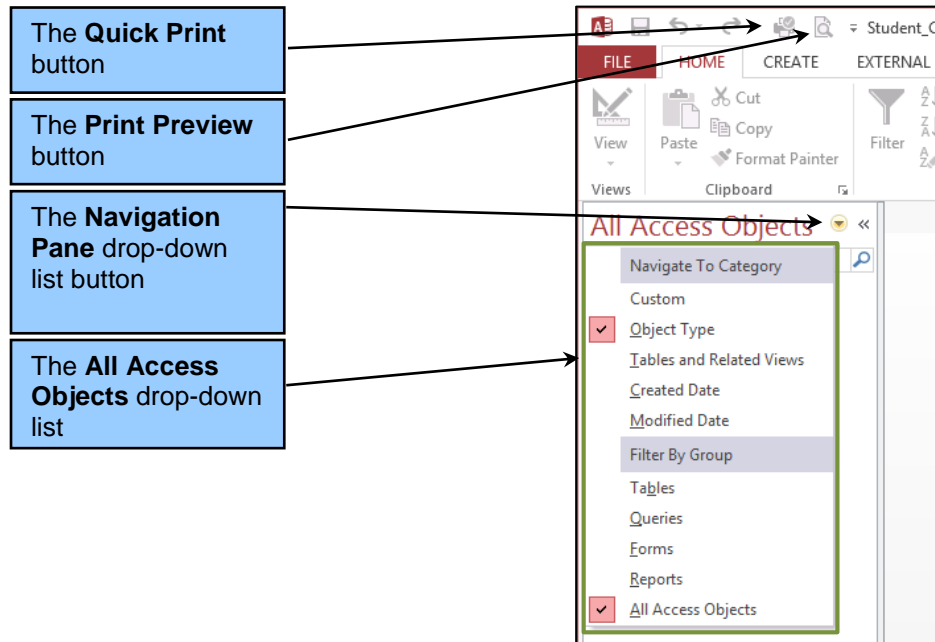


Figure A-8 — The Navigation Pane Drop-Down List

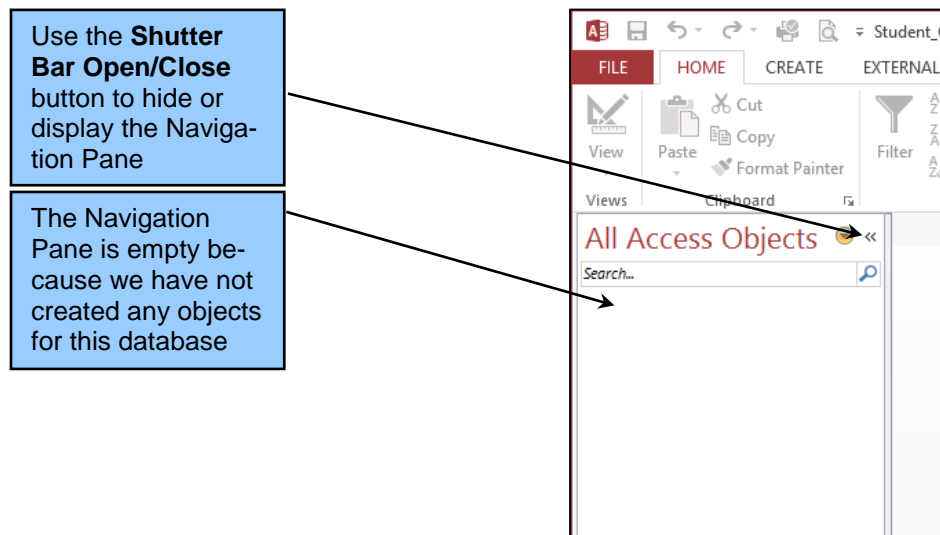


Figure A-9 — The Empty Navigation Pane

way you save Microsoft Word documents and Microsoft Excel workbooks. You can simply close a database, knowing that Microsoft Access has already saved all critical changes since you opened it.

Closing a Database and Exiting Microsoft Access

1. Click the **Close** button. The database closes, and you exit the Microsoft Access program.

BY THE WAY

Instead of clicking the Close button, you can close just the database while leaving Microsoft Access open by selecting **File | Close**.

How Do I Open an Existing Microsoft Access Database?

When we open an existing database, Microsoft Access 2013 gives us the option of using Microsoft Access security options to shut down certain Microsoft Access 2013 features in a database to protect ourselves against harm not only from viruses, but also from other possible problems. Unfortunately, the Microsoft Access 2013 security options also shut down significant and needed operational features of Microsoft Access. Therefore, we should normally enable the features that the Microsoft Access 2013 security warning warns us about when we open an existing database.

Opening a Recently Opened Microsoft Access Database

1. On the Windows 8 Start screen select the **Access 2013** app tile, as shown in Figure A-2 (If you are using Windows XP, Windows Vista or Windows 7, then select **Start | All Programs | Microsoft Office | Microsoft Access 2013**).
 2. The Microsoft Access 2013 window appears, as shown in Figure A-10. Note that the Microsoft Access 2013 window opens with the Access start screen displayed, which shows (1) a list of recently opened database files, (2) the **Open Other Files** button (used to browse for other database files, and (3) a set of Microsoft Access database templates available to the user.
 3. Note that the database file *Student_Class_Grade.accdb* is listed in the **Recent** list.
 4. Note that if the database has been used very recently, it will always be available in the Recent list. Otherwise, we will need to click the **File | Open** command to see it in the Recent page of that window. At this point, however, we can open the database by clicking on *either* entry. Because the Access start screen is open, click the **Student_Class_Grade.accdb** filename in the **Recent** list to open the database.
 5. A **Security Warning** bar appears with the database, as shown in Figure A-11.
 6. At this point, we have the option of clicking the Security Warning bar's **Click for more details** link, which will display a detailed version of the warning together with security options. However, for our purposes in this text, we simply need to enable the active content, so click the **Enable Content** button.
 - **NOTE:** At some point, you should select the **Click for more details** link, and explore the security settings available.
 - **NOTE:** In Microsoft Access 2007, the Security Warning bar appeared every time the database was re-opened (although only from a *non-trusted location* — see the discussion of *trusted locations* in the Microsoft Access 2013 documentation). In Microsoft Access 2013, the Security Warning bar is only displayed the first time you re-open a database, and your choice of options is remembered from that point on.
 - **NOTE:** You can also open an existing Microsoft Access 2013 by double-clicking on the file name in Microsoft Windows Explorer.
-

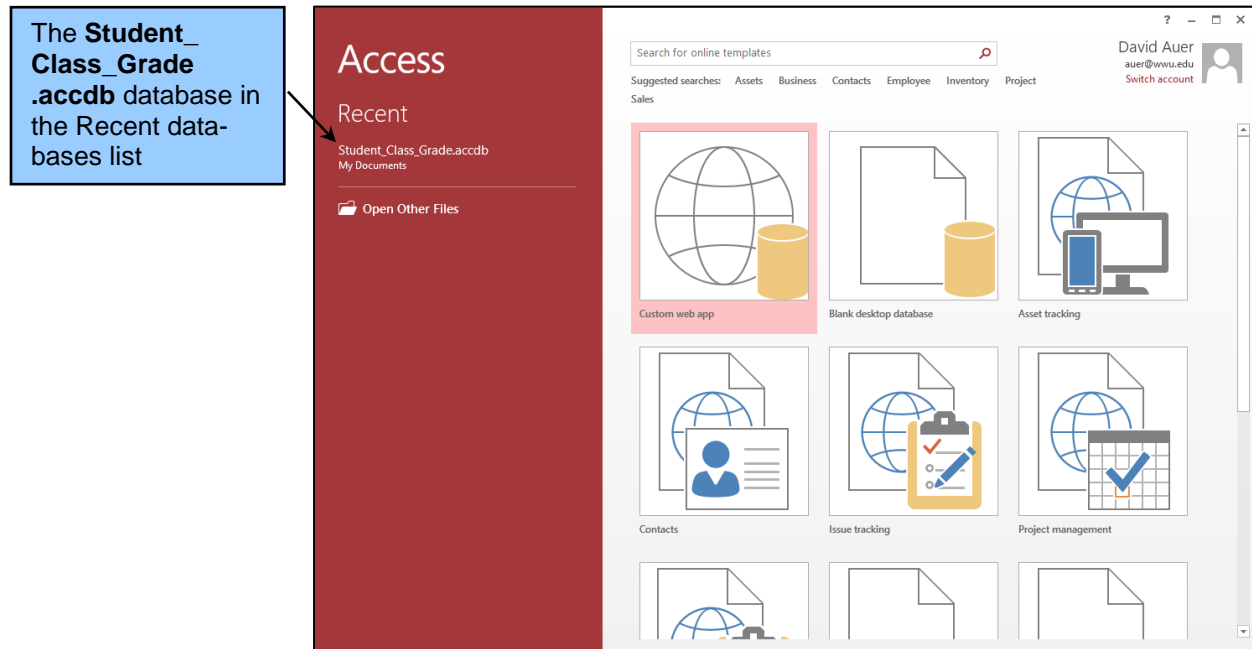


Figure A-10 — The Microsoft Access Start Screen and Recent List

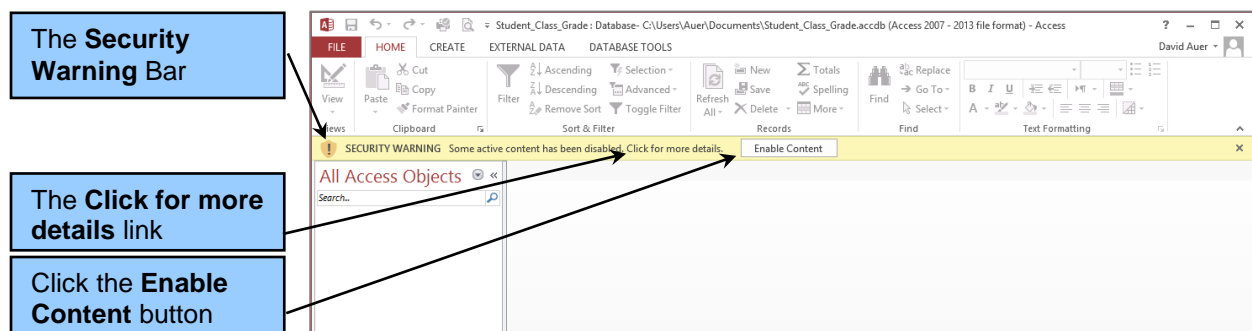


Figure A-11 — The Security Warning Bar

How Do I Create Microsoft Access Database Tables?

Now we need to create the tables in the Student-Class-Grade database—STUDENT, CLASS, and GRADE. However, because foreign key columns need their values to already exist in the primary keys of the tables being linked to, we should create our tables (and add data to our tables) in a specific order: we must create and populate (fill with data) STUDENT and CLASS first, and then create and populate GRADE (which has the foreign keys linking to STUDENT and CLASS). We will create the STUDENT table first.

The STUDENT table will contain the columns and characteristics shown in the table in Figure A-12. The column characteristics are type, key, required, and remarks.

Column Name	Type	Key	Required	Remarks
StudentNumber	AutoNumber	Primary Key	Yes	Surrogate Key
LastName	Text (25)	No	Yes	
FirstName	Text (25)	No	Yes	
EmailAddress	Text (100)	No	No	

Figure A-12 — Column Characteristics for the STUDENT Table

Type refers to the kind of data the column will store. Some possible Microsoft Access 2013 data types are shown in Figure A-13. For STUDENT, most data are being stored as **short text** data (also commonly called **character** data), which means we can enter strings of letters, numbers, and symbols (a space is considered a symbol).

Figure A-13 — Microsoft Access 2013 Data Types

Data Type Name	Type of Data	Size
Short Text	Characters and numbers (Formerly "Text")	Maximum 255 characters
Long Text	Characters and numbers (Formerly "Memo")	Maximum 65,535 characters
Number	Numeric Data	Varies with number type
Date/Time	Dates and time from the year 100 to the year 9999	Stored as 8-byte double-precision integers
Currency	Numbers with decimal places	One to four decimal places
AutoNumber	A unique sequential number	Incremented by one each time
Yes/No	Fields that can only contain two values	Yes/No, On/Off, True/False
OLE Object	An object embedded in or linked to an Access table	Maximum 1 GB
Hyperlink	A hyperlink address	Maximum 2,048 characters in each of the three parts of the hyperlink address
Attachment	Any supported file type can be attached to a record	Independent of Access
Calculated	Calculates a values from data in other field	Dependent upon data used in calculation
Lookup Wizard	Creates a multivalued field based on an Access table or query	Dependent upon data used in lookup

The number after the word *Text* in Figure A-12 indicates how many characters can be stored in the column. For example, student last names may be up to 25 characters long. The only **number**, or **numeric**, data column in the STUDENT table is StudentNumber, which is listed as **AutoNumber**. This indicates that Microsoft Access will automatically provide a sequential number for this column for each new student that is added to the table.

Key refers to table identification functions assigned to a column. These are described in detail in Chapter 3. At this point, you simply need to know that a primary key is a column value used to identify each row, and, therefore, the values in this column must be unique. This is the reason for using the AutoNumber data type, which automatically assigns a unique number to each row in the table as the row is created.

Required refers to whether the column must have a data value. If it must, a value must be present in the column. If not, the column may be blank. Note that because StudentNumber is a primary key used to identify each row, it *must* have a value.

Remarks contains comments about the column or how it is used. For STUDENT, the only comment is that StudentNumber is a *surrogate key*. Surrogate keys are discussed in Chapter 3. At this point, you simply need to know that a **surrogate key** is usually a set of computer-generated unique numbers used to identify rows in a table (that is, a primary key). This is done by using the Microsoft Access AutoNumber data type.

Creating the STUDENT Table

1. Click the **CREATE** command tab to display the **CREATE** command groups.
2. Click the **Table Design** button, as shown in Figure A-14. The **Table1** tabbed document window is displayed in **Design** view, as shown in Figure A-15. Note that along with the **Table1** window a contextual tab grouping named **TABLE TOOLS** is displayed and that this tab grouping adds a new command tab named **DESIGN** to the set of command tabs displayed.

■ **NOTE:** It seems like now would be a good time to name the new table STUDENT. With Microsoft Access, however, you don't name a table until you save it the first time, and you can't save a table until you have at least one column defined. So, we will define the columns, and then we will save and name the table. If you want, save the table after you've defined just one column.

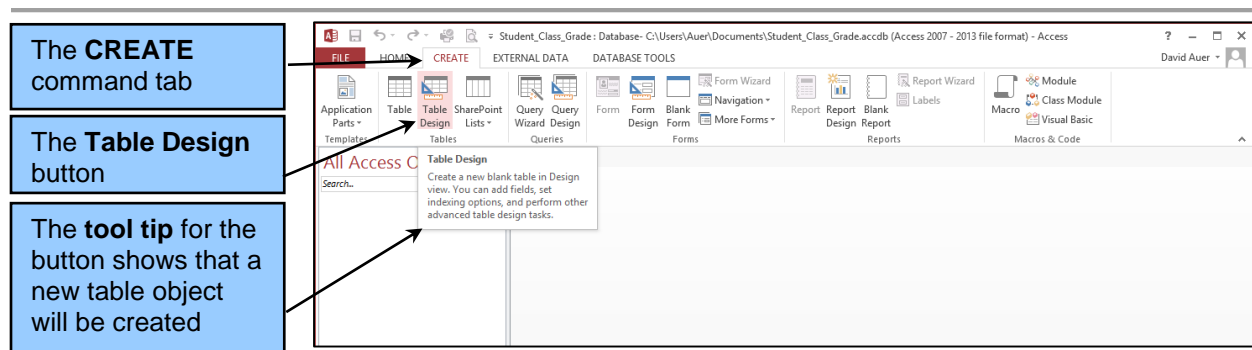


Figure A-14 — The Table Design Button

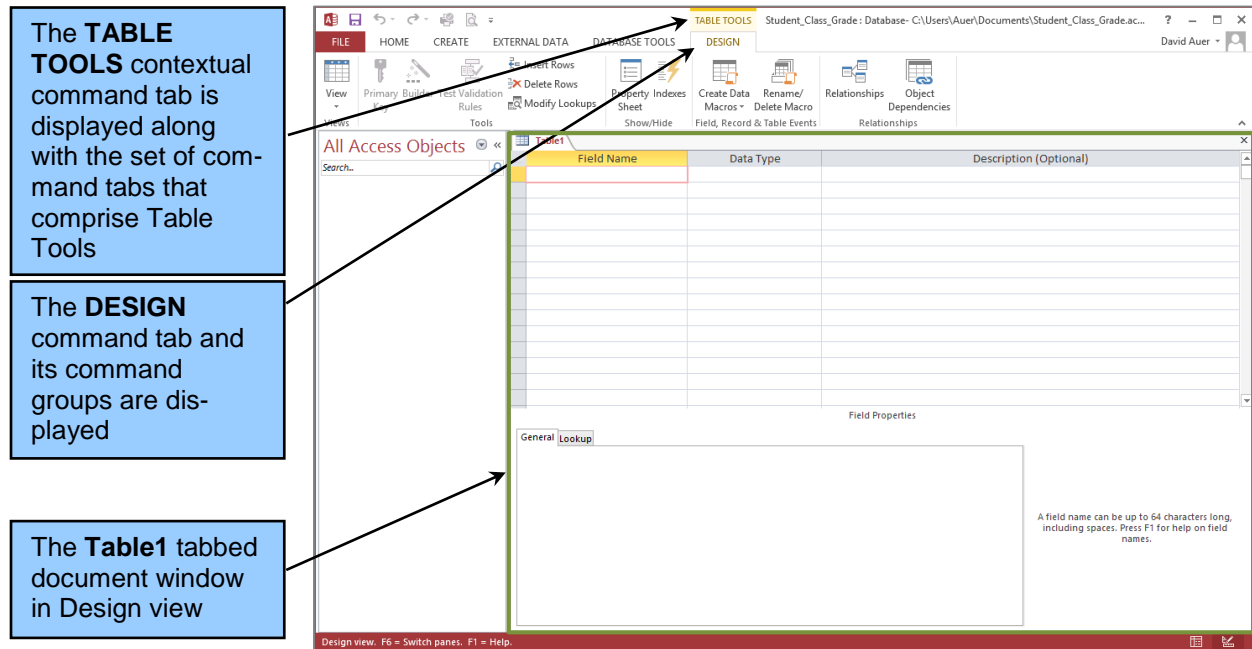


Figure A-15 — The Table1 Tabbed Document Window

3. In the **Field Name** column text box of the first line, type the column name **StudentNumber** and then press the **Tab** key to move to the **Data Type** column. (You can also click the Data Type column to select it.)
 - **NOTE:** The terms *column* and *field* are considered synonyms in database work. The term *attribute* is also considered to be equivalent to these two words.
4. Select the **AutoNumber** data type for StudentNumber from the **Data Type** drop-down list, as shown in Figure A-16.
5. If you like, an optional comment may be stored in the Description column. To do so, move to the Description column by pressing the **Tab** key or clicking in the **Description** text box. Type the text **Surrogate key for STUDENT** and then press the **Tab** key to move to the next row. The **Table1** tabbed document window now looks as shown in Figure A-17.
 - **NOTE:** The Remarks column in the set of database column characteristics shown in Figure A-12 is *not* the same as the table Description column shown in Figure A-17. Be careful not to confuse them. The Remarks column is used to record technical data, such as facts about table keys and data default values that are necessary for building the table structure. The Description column is used to describe to the user the data stored in that field so that the user understands the intended use of the field.

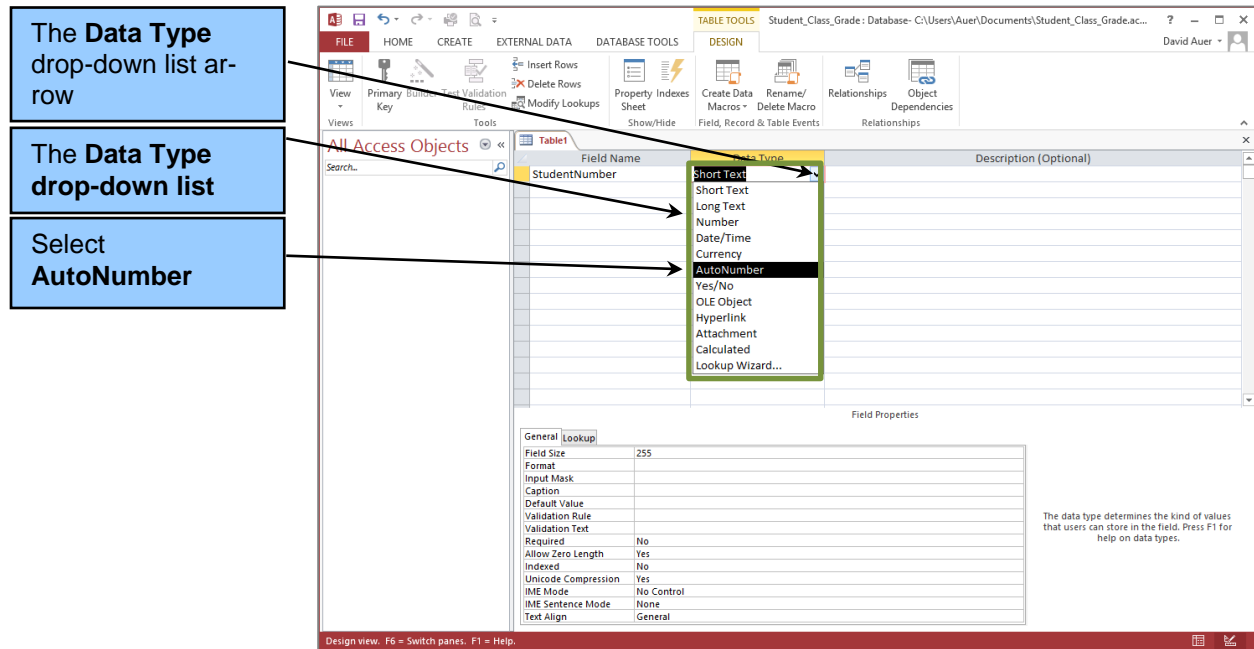


Figure A-16 — Selecting the Data Type

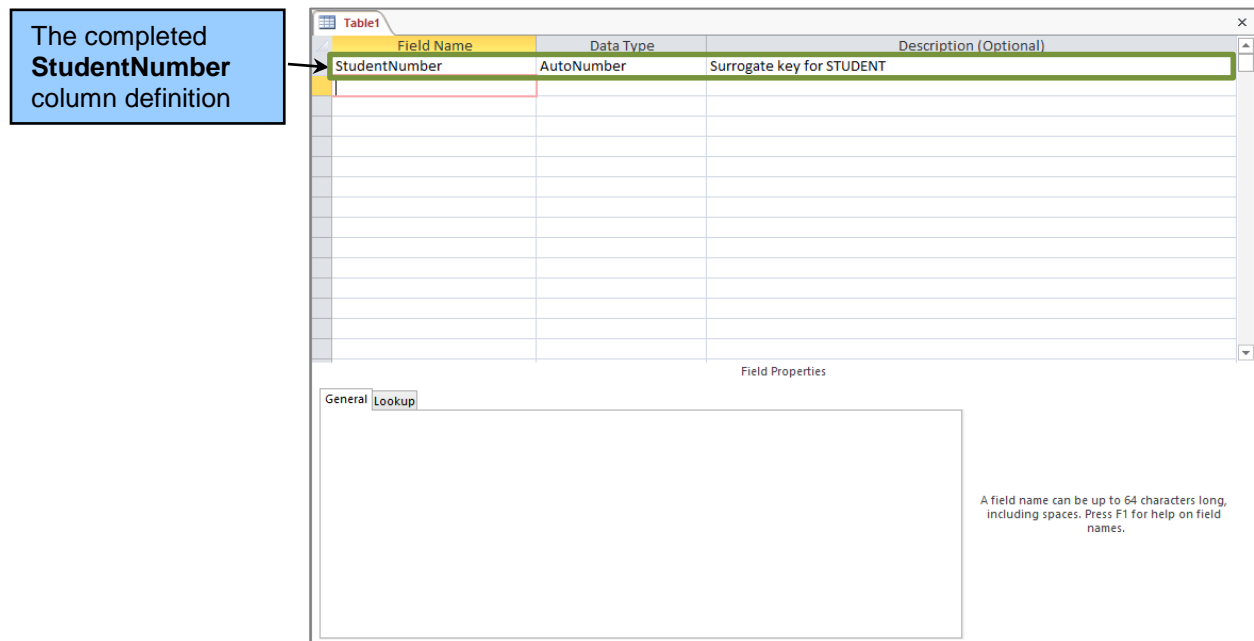


Figure A-17 — The Completed StudentNumber Column Definition

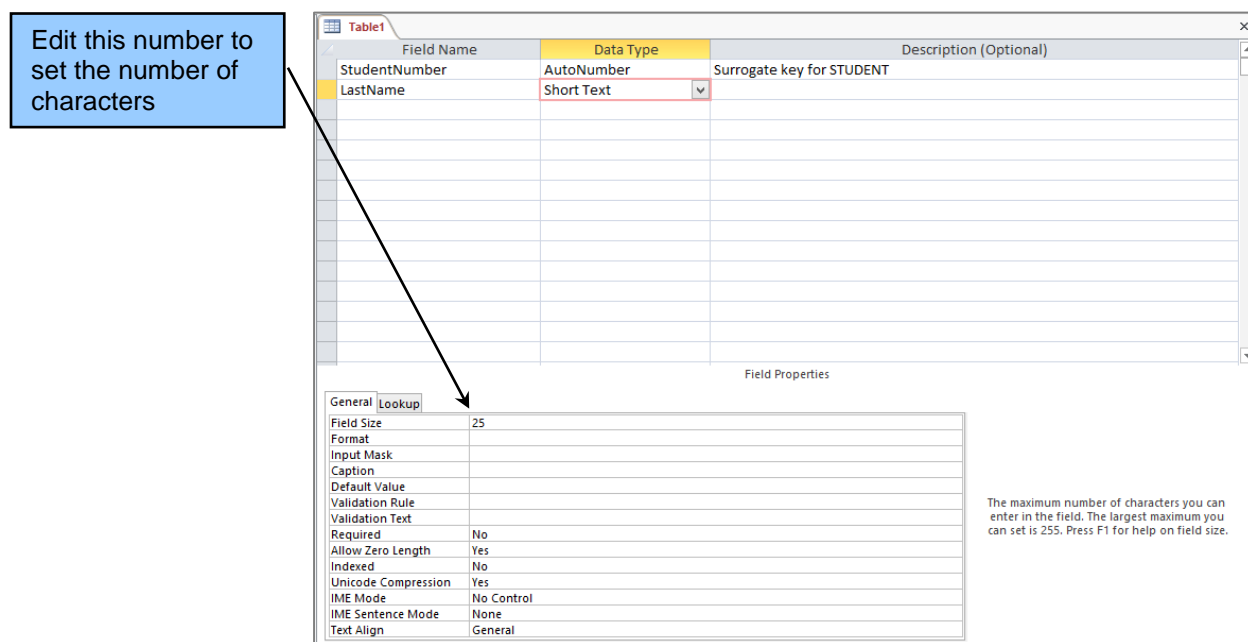


Figure A-18 — Editing the Text Field Size

6. The other columns of the STUDENT table are created using the sequence described in steps 3 through 5.
 - **NOTE:** See Figure A-20 for the full set of Description column entries for this table.
7. To set the number of characters in text columns, edit the Short Text data type **Field Size** text box as shown in Figure A-16. The default value for Field Size is 255, and the maximum value is 255.
8. To make a column required, click anywhere in the column Data Type Required property text box to display the **Required** property drop-down list arrow button and then click the button to display the Required property drop-down list, as shown in Figure A-19.
9. Select **Yes** from the Required property drop-down list. The default is No (not required), and Yes must be selected to make the column required.²

Now we need to set a primary key for the STUDENT table. According to Figure A-12, we need to use the StudentNumber column as the primary key for this table.

² Microsoft Access has an additional Data Type property named Allow Zero Length. This property confounds the settings necessary to truly match the SQL constraint NOT NULL discussed in Chapter 7. However, the discussion of Allow Zero Length is beyond the scope of this book. See the Microsoft Access Help system for more information.

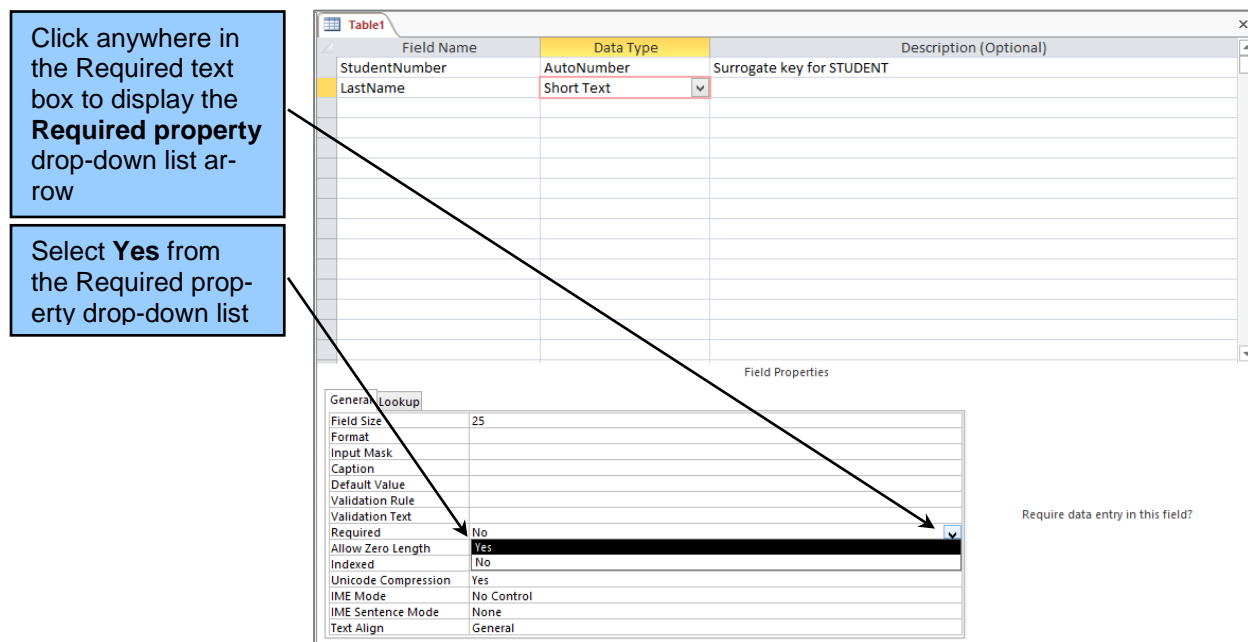


Figure A-19 — Setting the Column Required Property Value

Setting the STUDENT Table Primary Key

1. Move the mouse pointer to the **row selector column** of the row containing the StudentNumber properties, as shown in Figure A-20, and click to select the row.
2. Click the **Primary Key** button in the Tools group of the Design tab, as shown in Figure A-21. StudentNumber is selected as the primary key for the STUDENT table.

We have finished building the STUDENT table. Now we need to name the table, save it, and close it.

Naming, Saving, and Closing the STUDENT Table

1. To name and save the STUDENT table, click the **Save** button in the Quick Access Toolbar. The **Save As** dialog box appears, as shown in Figure A-22.
2. Type the table name **STUDENT** into the **Save As** dialog box's Table Name text box and then click the **OK** button. The table is named and saved. The table name STUDENT now appears on the document tab, and the STUDENT table object is displayed in the Navigation Pane, as shown in Figure A-23.

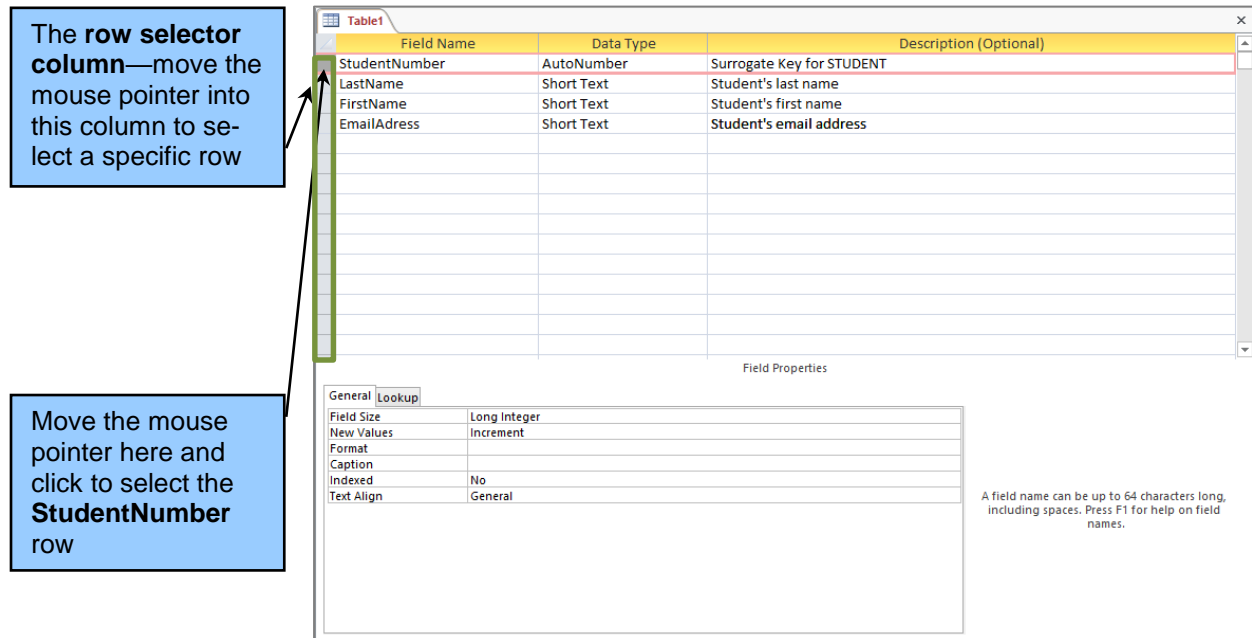


Figure A-20 — Selecting the StudentNumber Row

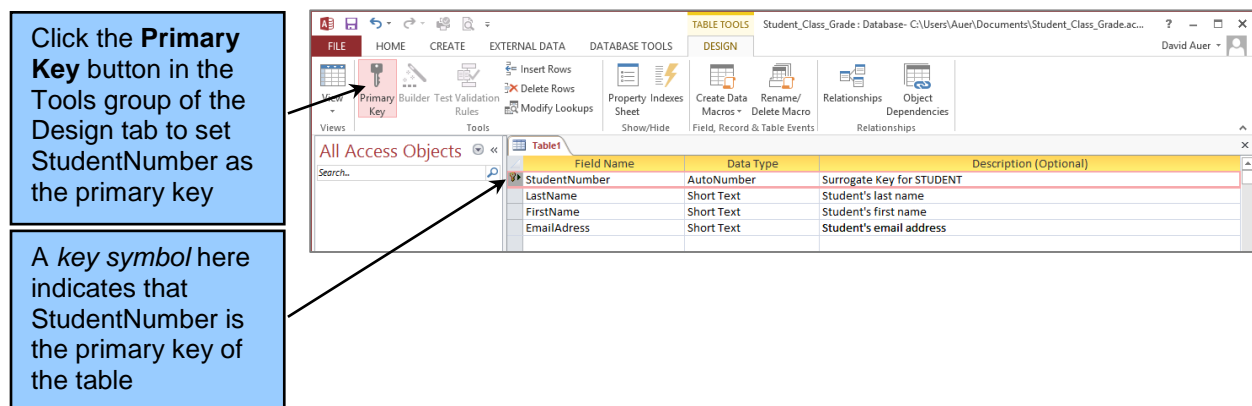


Figure A-21 — Setting the Primary Key

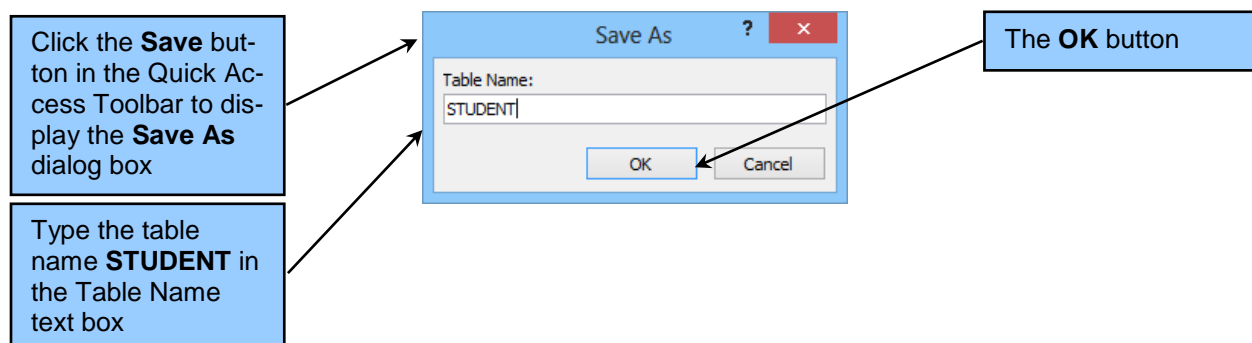


Figure A-22 — Naming and Saving the STUDENT Table

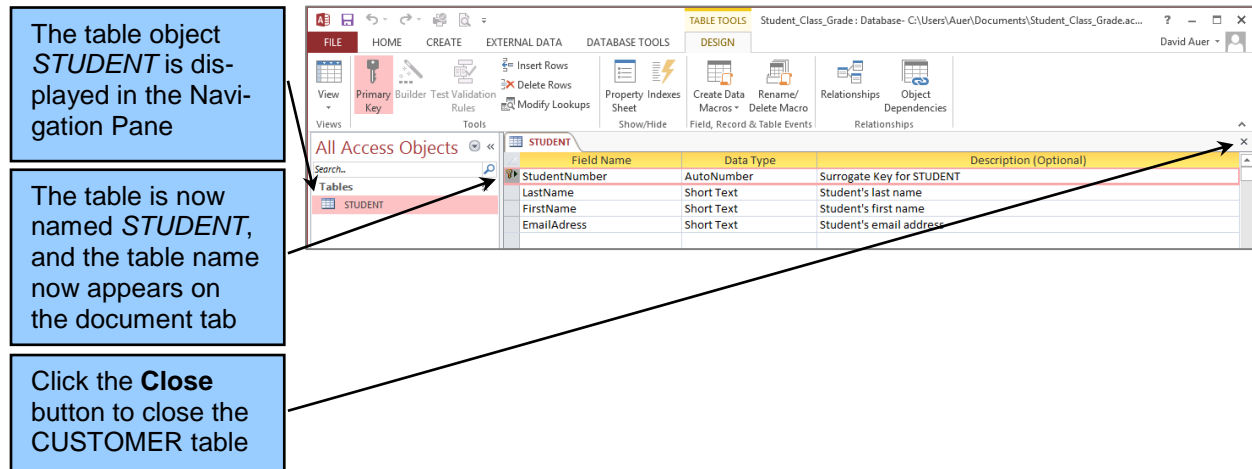


Figure A-23 — The Saved and Opened STUDENT Table

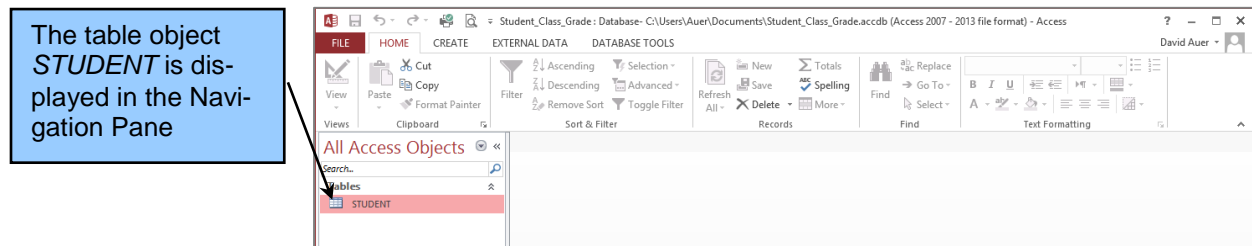


Figure A-24 — The STUDENT Table Object in the Navigation Pane

- To close the STUDENT table, click the **Close** button in the upper-right corner of the tabbed documents window. After the table is closed, the STUDENT table object remains displayed in the Navigation Pane, as shown in Figure A-24.

At this point, we need to build the CLASS and GRADE tables. The CLASS table will have the column characteristics shown in Figure A-25, and the GRADE table will have the column characteristics shown in Figure A-26. Using steps similar to those you used for the STUDENT table, create the CLASS and GRADE tables. Note that GRADE has a **composite primary key**, and in order to designate this key you must select the row representing StudentNumber, as discussed earlier, and then CTRL-Click (hold down the CTRL key and then click) on the row representing ClassNumber so that both are highlighted when you click the Primary Key button.

Column Name	Type	Key	Required	Remarks
ClassNumber	Number	Primary Key	Yes	Long Integer
ClassName	Text (25)	No	Yes	
Term	Text (12)	No	Yes	
Section	Number	No	Yes	Integer

Figure A-25 — Column Characteristics for the CLASS Table

Column Name	Type	Key	Required	Remarks
StudentNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
ClassNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
Grade	Number	No	Yes	Decimal, Fixed, Scale = 2, Decimal Places = 1

Figure A-26 — Column Characteristics for the GRADE Table

How Do I Insert Data into Tables Using the Datasheet View?

There are three commonly used methods for adding data to a table. First, we can use a table as a **datasheet**, which is visually similar to and works like an Excel worksheet. When we do this, the table is in **Datasheet view**, and we enter the data cell by cell. Second, we can build a **data entry form** for the table and then use the form to add data. Third, we can use SQL to insert data. This section covers entering data into a datasheet. We will discuss forms later in this Appendix, and we will use SQL to insert data in Chapters 4 and 7.

In Microsoft Access 2013, we can also use Datasheet view to create and modify table characteristics: When we open a table in Datasheet view, the Table Tools contextual tab includes a Datasheet command tab and ribbon with tools to do this. We do *not* recommend this and prefer to use Design view, as previously discussed in this section, for creating and modifying table structures.

Of course, before we can use either method for creating and modifying table structures, we need to put some data into the table. Figure A-27 shows some STUDENT data.

StudentNumber	LastName	FirstName	EmailAddress
1	Cooke	Sam	Sam.Cooke@OurU.edu
2	Lau	Marcia	Marcia.Lau@OurU.edu
3	Harris	Lou	Lou.Harris@OurU.edu
4	Greene	Grace	Grace.Greene@OurU.edu

Figure A-27 — Sample STUDENT Data

Adding Data to the STUDENT Table in Datasheet View

1. In the Navigation Pane, double-click the **STUDENT** table object. The STUDENT table window appears in a tabbed document window in Datasheet view, as shown in Figure A-28. Note that if some columns on the right side of the datasheet do not appear in the window, but you can access them by scrolling or minimizing the Navigation Pane.

■ **NOTE:** As in a worksheet, the intersection of a row and column is called a **cell** in a datasheet.

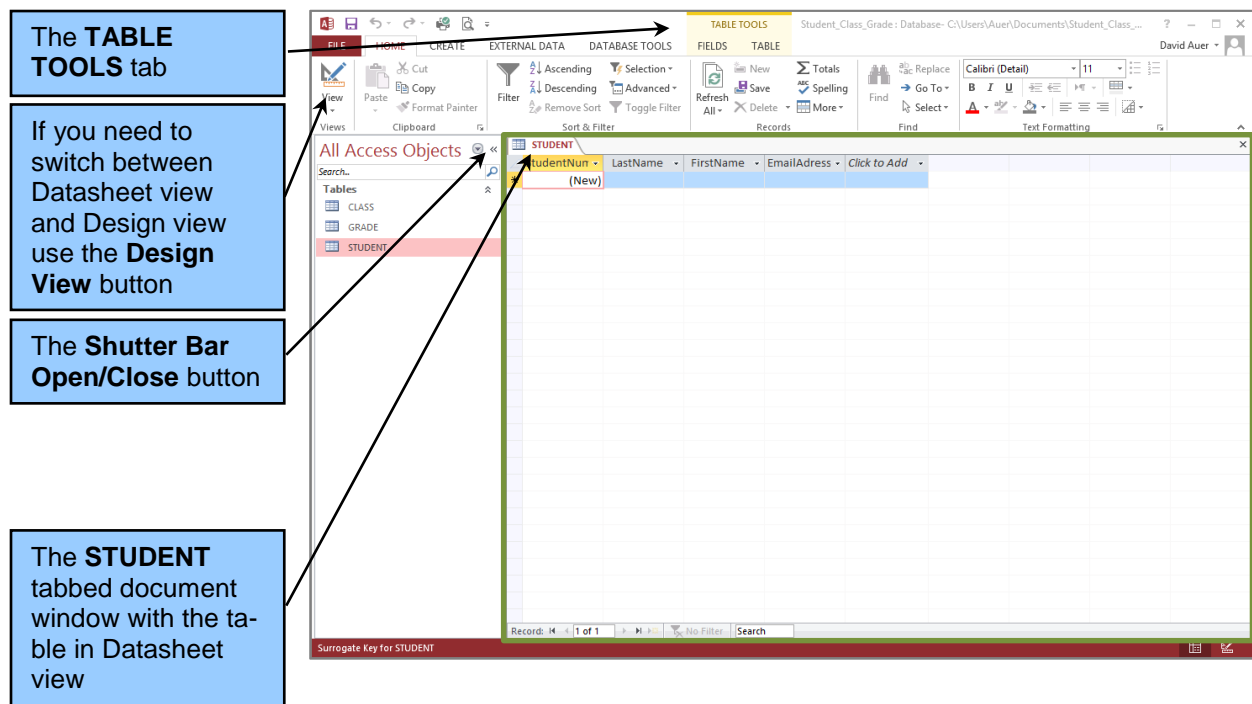


Figure A-28 — The STUDENT table in Datasheet View

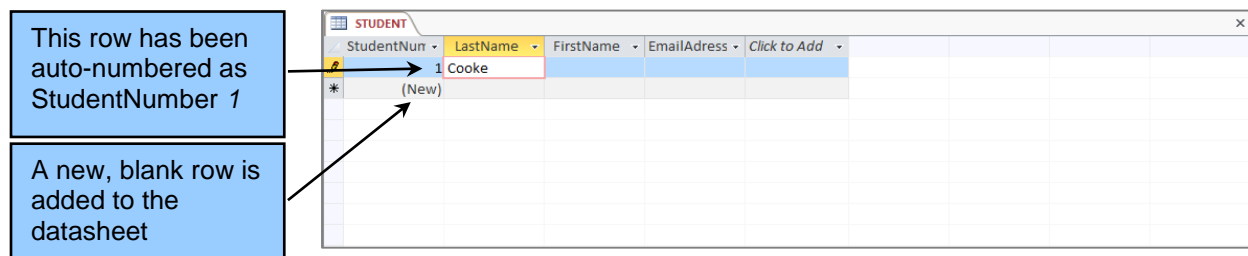


Figure A-29 — Entering Data Values for Sam Cooke

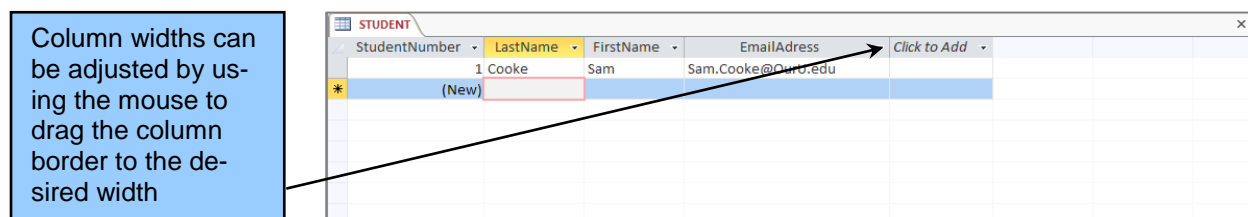


Figure A-30 — The Completed Row of Data Values

2. Click the **STUDENT** document tab to select the STUDENT table in Datasheet view.
 3. Click the cell in the StudentNumber column with the phrase **(New)** in it to select that cell in the new row of the STUDENT datasheet.
 4. Press the **Tab** key to move to the LastName cell in the new row of the STUDENT datasheet. For student Sam Cooke, type **Cooke** in the LastName cell. Note that as soon as you do this, the AutoNumber function puts the number 1 in the StudentNumber cell, and a new row is added to the datasheet, as shown in Figure A-29.
 5. Using the **Tab** key to move from one column to another in the STUDENT datasheet, enter the rest of the data values for Sam Cooke.
 6. The final result is shown in Figure A-30. Note that the width of the Email column was expanded using the mouse to move the border of the column—just as you do in an Excel worksheet.
- **NOTE:** If you make a mistake and need to return to a cell, click the cell to select it, and you go into Edit mode. Alternatively, you can use **Shift-Tab** to move to the right in the datasheet and then press **F2** to edit the contents of the cell.
 - **NOTE:** Remember that LastName and FirstName *require* a data value. You will not be able to move to another row or close the table window until you have some value in each of these cells.
 - **NOTE:** Figure A-30 shows a column labeled *Click to Add* to the right of the Email column. This is a table tool in Datasheet view that you can use to create or modify table structures. We do not recommend using these tools; we prefer to use Design view instead!
 - **NOTE:** Altering the width of a column in datasheet view does not change the amount of data stored in the database – it only changes the way the data is presented in datasheet view.

7. Use the **Tab** key to move to the next row of the STUDENT datasheet and enter the data for Marcia Lau.
8. Enter the data for Lou Harris.
9. Enter the data for Grace Greene.
10. If necessary, adjust the datasheet column widths so that you can see the contents of the datasheet in one screen. The final result is shown in Figure A-31.
11. Click the **Close** button in the upper-right corner of the document window to close the STUDENT datasheet. If a dialog box appears asking if you want to save the changes you made to the layout (column widths), click the **Yes** button.

At this point, we need to enter data into the CLASS table, but *not* into the GRADE table. The data in the GRADE table will be entered only after we have created the relationships between the tables. The data for the CLASS table are shown in Figure A-32. Enter this data using the CLASS table in Datasheet view.

Modifying and Deleting Data in Tables in the Datasheet View

After data is entered into a table, you can modify or change it by editing the data values in the Datasheet view. You can also delete rows of data in the Datasheet view. For information on how to do this, see the Microsoft Access 2013 documentation in the Help system.

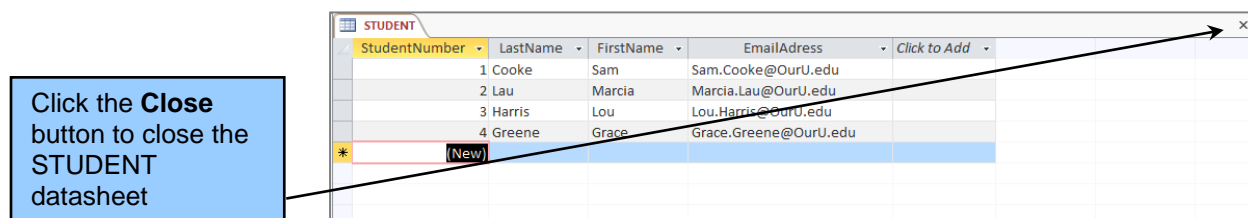


Figure A-31 — The Completed STUDENT Datasheet

ClassNumber	ClassName	Term	Section
10	CHEM 101	2014-Fall	1
20	CHEM 101	2014-Fall	2
30	CHEM 101	2015-Spring	1
40	ACCT 101	2014-Fall	1
50	ACCT 102	2015-Spring	1

Figure A-32 — Sample CLASS Data

How Do I Create Relationships Between Tables?

In Microsoft Access, you build relationships between tables by using the **Relationships window**, which you access by using the **Tools | Relationships** command.

Creating the Relationship Between the STUDENT and GRADE Tables

1. Click the **DATABASE TOOLS** tab to display the DATABASE TOOLS command groups, as shown in Figure A-33.
2. Click the **Relationships** button in the Show/Hide group. As shown in Figure A-34, the Relationships tabbed document window appears, together with the Show Table dialog box. Note that along with the Relationships window, a contextual tab named **RELATIONSHIP TOOLS** is displayed and that this tab adds a new command tab named **DESIGN** to the set of command tabs displayed.
3. In the Show Table dialog box, click the **STUDENT** table to select it. Click the **Add** button to add STUDENT to the Relationships window.
4. In the Show Table dialog box, click the **GRADE** table to select it. Click the **Add** button to add GRADE to the Relationships window.
5. In the Show Table dialog box, click the **Close** button to close the dialog box.
6. Rearrange and resize the table objects in the Relationships window using standard Windows drag-and-drop techniques. Rearrange the STUDENT and GRADE table objects until they appear as shown in Figure A-35. Now we are ready to create the relationship between the tables.

■ **NOTE:** A formal description of how to create a relationship between two tables is “In the Relationships window, drag a primary key column and drop it on top of the corresponding foreign key column.” It is easier to understand this after you have actually done it.

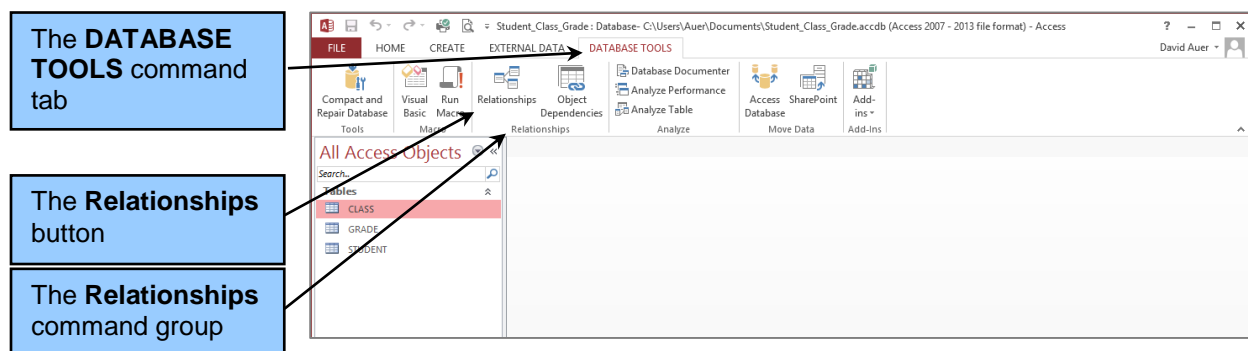


Figure A-33 — The DATABASE TOOLS Command Tab

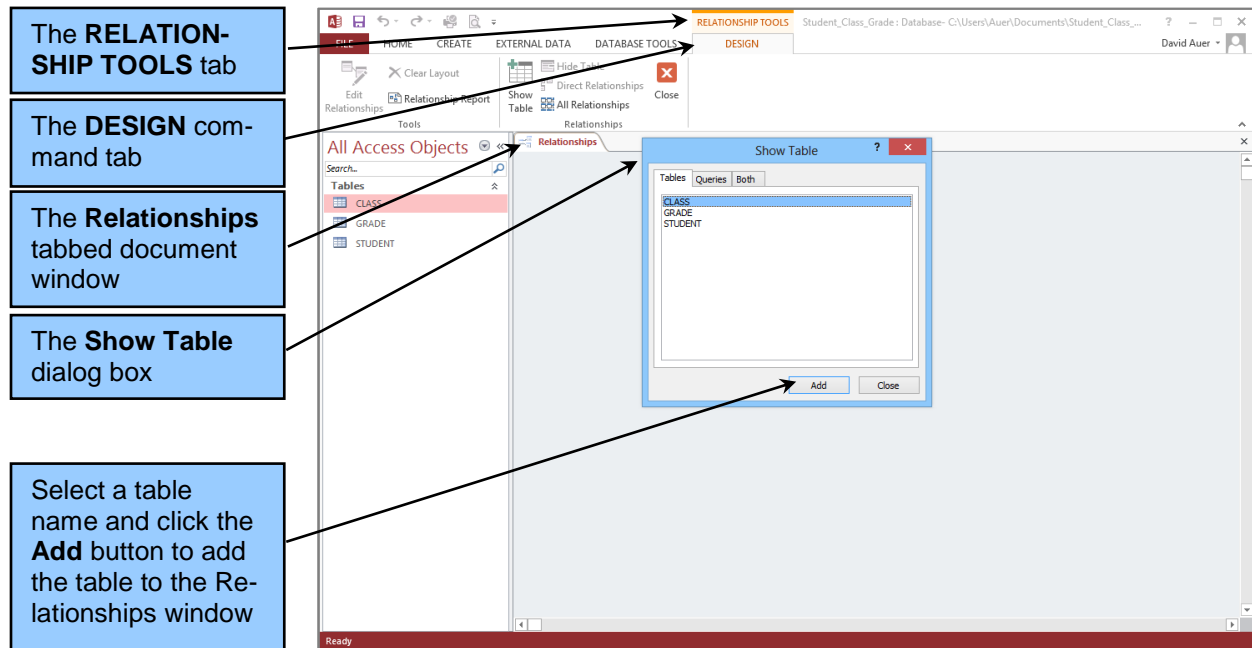


Figure A-34 — The Relationships Window

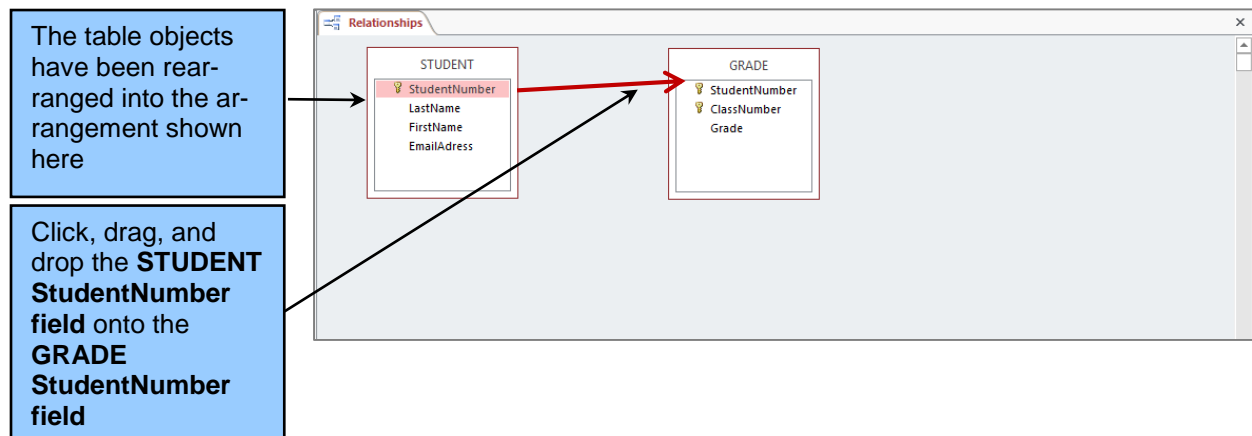


Figure A-35 — The Table Objects in the Relationships Window

7. Click and hold the **column name StudentNumber** in the **STUDENT** table and then drag it over the **column name StudentNumber** in the **GRADE** table. The Edit Relationships dialog box appears, as shown in Figure A-36.
 - **NOTE:** In **STUDENT**, **StudentNumber** is the primary key, and in **GRADE**, **StudentNumber** is the foreign key.
8. Click the **Enforce Referential Integrity** check box.

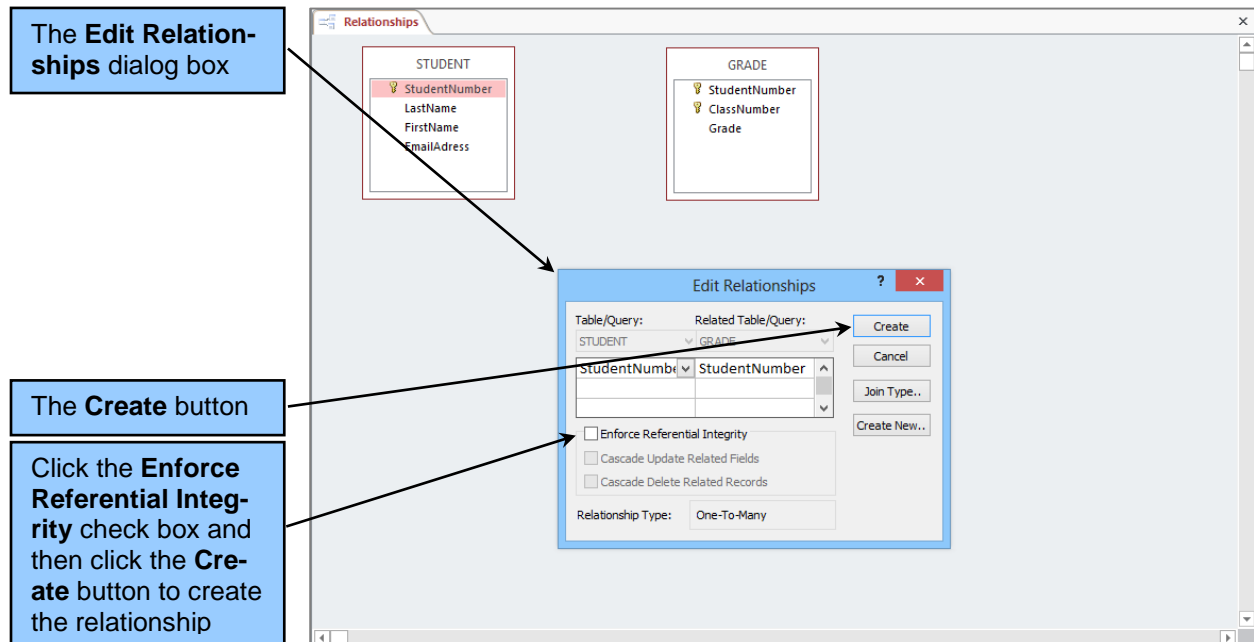


Figure A-36 — The Edit Relationships Dialog Box

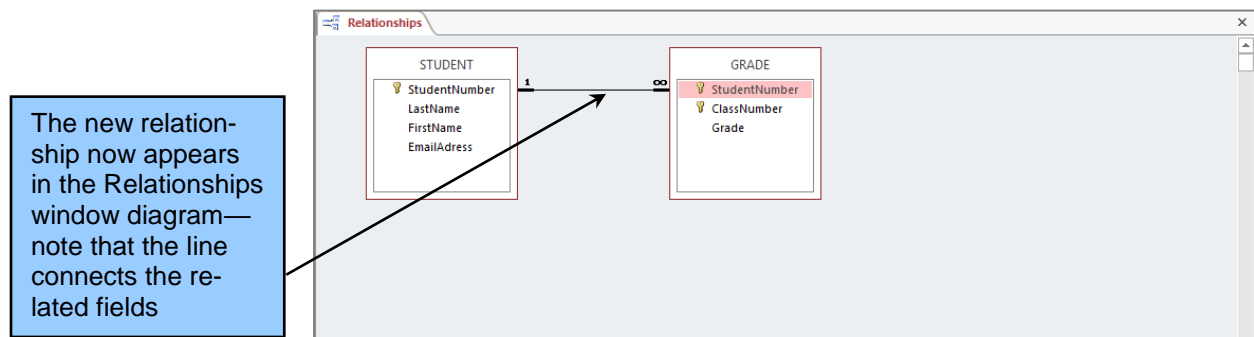


Figure A-37 — The Completed Relationship

9. Click the **Create** button to create the relationship between STUDENT and GRADE. The relationship between the tables now appears in the Relationships window, as shown in Figure A-37.
10. To close the Relationships window, click the **Close** button in the upper-right corner of the document window. A Microsoft Access dialog box appears, asking whether you want to save changes to the layout of relationships. Click the **Yes** button to save the changes and close the window.

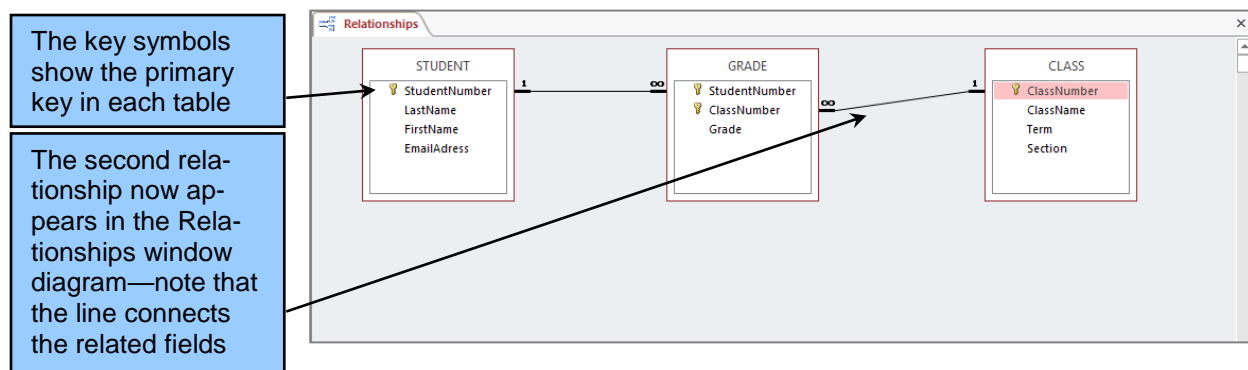


Figure A-38 — The Completed Relationships for the Student-Class-Grade Database

Now we need to repeat essentially the same steps and create the relationship between CLASS and GRADE. When we are done, the relationships appear as shown in Figure A-38.

At this point, referential integrity has been established between the foreign keys in GRADE and the corresponding primary keys in STUDENT and CLASS. This means that we cannot add a data value to StudentNumber or ClassNumber in GRADE unless the same value already exists in StudentNumber in STUDENT or ClassNumber in CLASS. This prevents us from inserting erroneous data into GRADE.

Now we can enter the GRADE data shown in Figure A-39 into the GRADE table. After this is done, we have completed creating and populating the Student-Class-Grade database.

StudentNumber	ClassNumber	Grade
1	10	3.7
1	40	3.5
2	20	3.7
3	30	3.1
4	40	3.0
4	50	3.5

Figure A-39 — Sample GRADE Data

How Do I Create and Run Microsoft Access 2013 Queries?

Once we have data in a database, we can use that data to extract information about the data. Basically, this means that we want to ask a question and get an answer based on the data. For example, using the Student-Class-Grade database, we could ask “Who has taken ACCT 101?” Given the data we are using, the answer is “Grace Greene.”

In Microsoft Access 2013, we ask the database a question by creating and running a Microsoft Access **query**. There are two query methods that we use in Microsoft Access. The first method is **Structured Query Language (SQL)**, which is mentioned in Chapter 1 and discussed in Chapter 2. SQL is text based, and, as the name implies, is designed (in large part, but not entirely as you will learn in Chapter 7) for querying databases. The second method is the Microsoft Access version of **Query by Example (QBE)**, which uses the Microsoft Access GUI to build queries. QBE is the Microsoft Access default, and we will discuss it here. We will examine how to use SQL in Microsoft Access in Chapter 2. It is important to note that Microsoft Access, unlike enterprise-class DBMS products, will allow you to save queries as part of the database structure. This is a function of the application development component of Microsoft Access, just as are forms and reports (which we will discuss later in this Appendix).

Queries are based on one or more tables (and in Microsoft Access can also be based on other existing, saved queries). To understand how Microsoft Access QBE works, we will use QBE to create a multi-table query. The query will provide student enrollment and final grade information for each class and is the same query that was used as the basis for the Class Grade Report in Figure 1-13.

Creating and Running a Microsoft Access QBE Query

1. Click the **CREATE** command tab to display the Create command groups, as shown in Figure A-40.
2. Click the **Query Design** button in the Queries command group on the CREATE command tab.
3. The Query1 tabbed document window is displayed in Design view, along with the Show Table dialog box, as shown in Figure A-41.

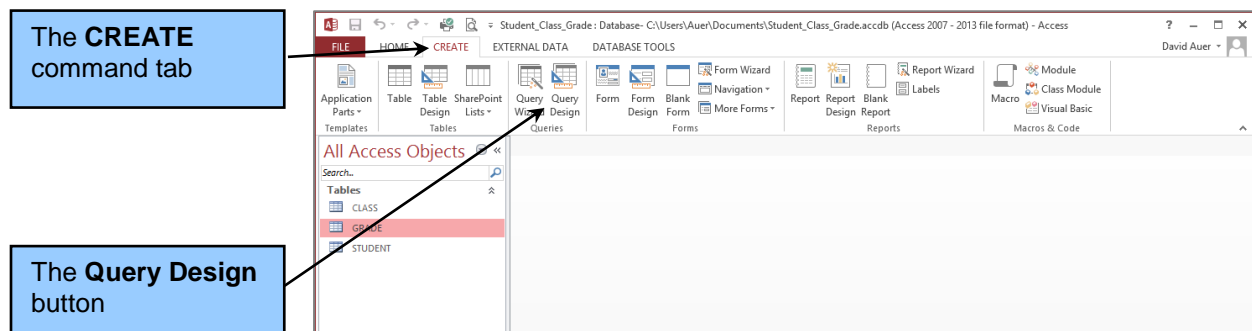


Figure A-40 — The CREATE Command Tab

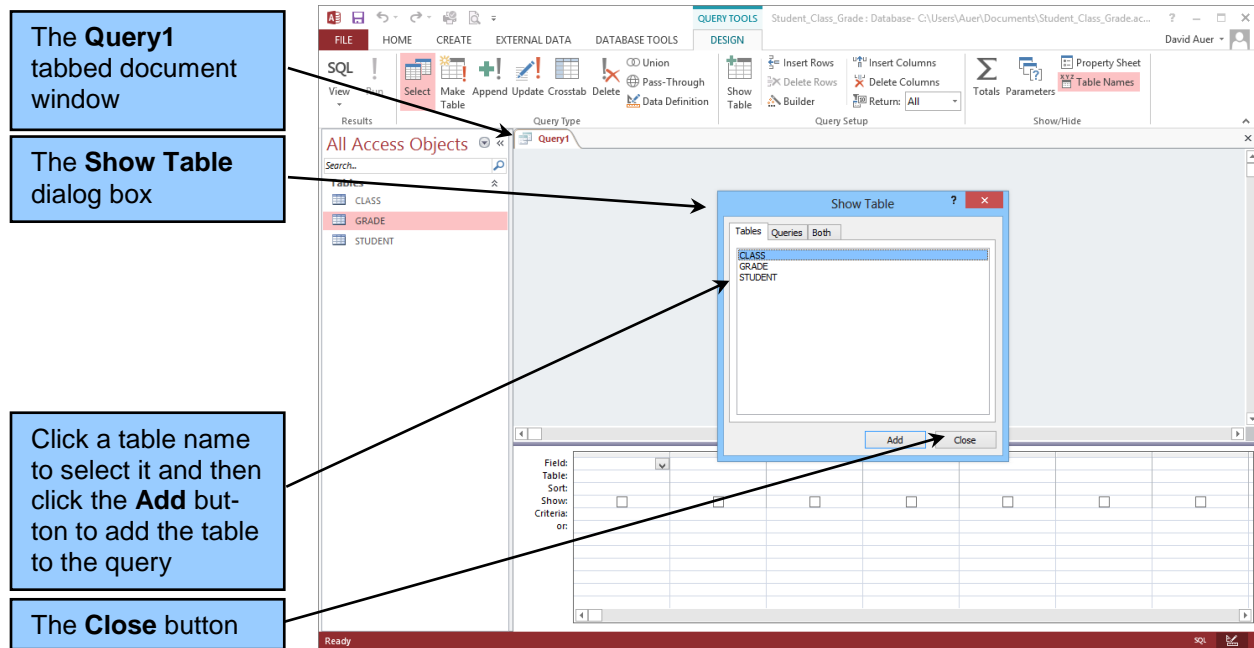


Figure A-41 — The Show Table Dialog Box

4. In the Show Table dialog box, click **CLASS** to select the CLASS table. Click the **Add** button to add the CLASS table to the query.
5. In the Show Table dialog box, click **GRADE** to select the GRADE table. Click the **Add** button to add the GRADE table to the query.
6. In the Show Table dialog box, click **STUDENT** to select the STUDENT table. Click the **Add** button to add the STUDENT table to the query.
7. Click the **Close** button to close the Show Table dialog box.
8. Rearrange and resize the query window objects in the Query1 query document window, using standard Windows drag-and-drop techniques, until they appear as shown in Figure A-42.
9. Note the elements of the Query1 window shown in Figure A-42: Tables and their associated set of columns—called a field list—that are included in the query are shown in the upper pane, and the columns (fields) actually included in the query are shown in the lower pane. For each included column (field), you can set whether this column's data appear in the results, how the data are sorted, and the criteria for selecting which rows of data will be shown. Note that the first entry in the table's field list is the asterisk (*), which has its standard SQL meaning of "all columns in the table."
10. We include columns in the query by dragging them from the table's field list to a field column in the lower pane. Click and drag **ClassNumber** in CLASS to the first field column, as shown in Figure A-43. Note that the column is entered as ClassNumber from the table CLASS.
11. Click the **Shutter Bar Open/Close** button to collapse the Navigation Pane—we will need the extra space to build the query.

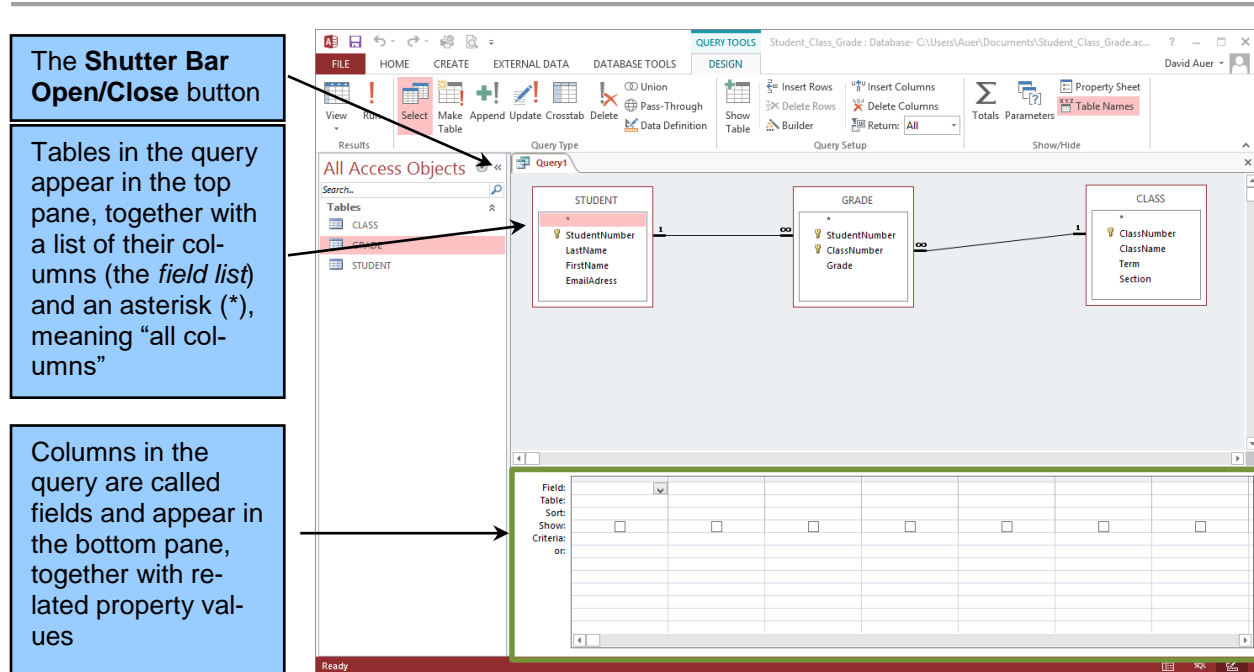


Figure A-42 — The QBE Query1 Query Window

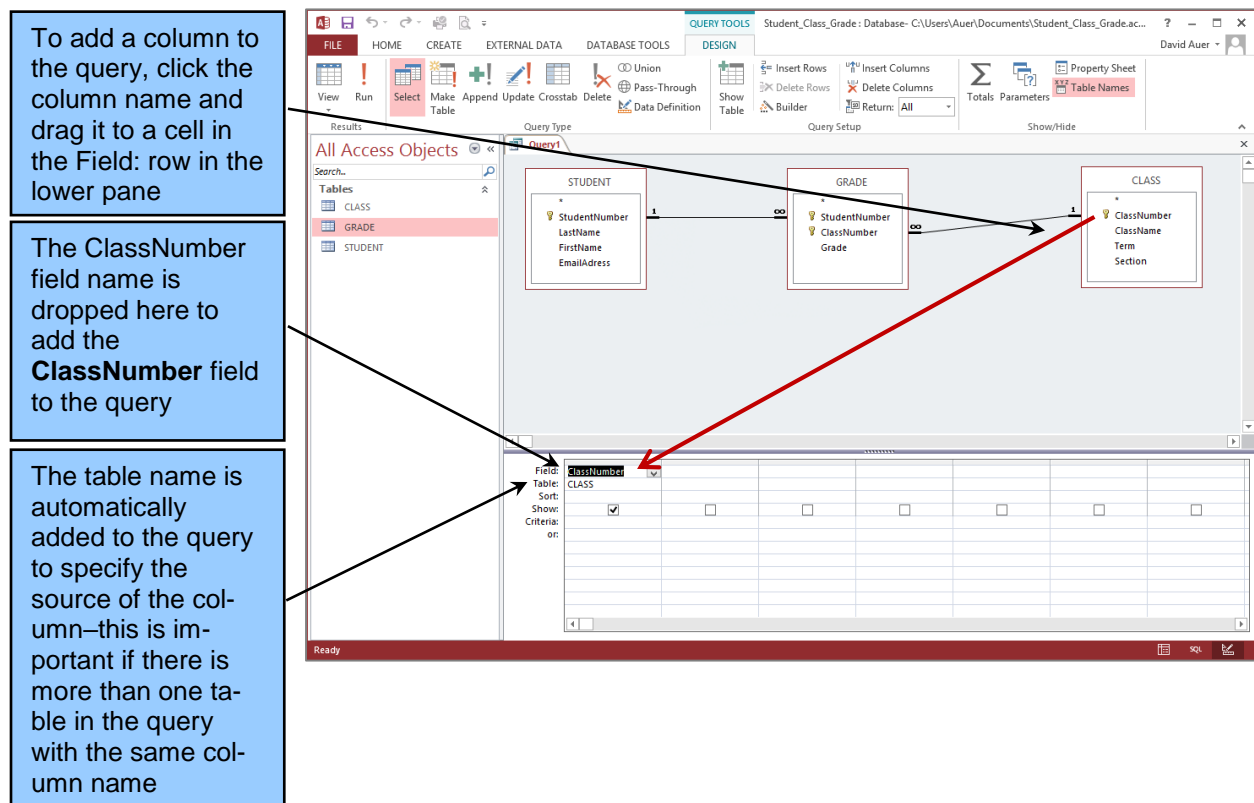


Figure A-43 — Adding Columns to the QBE Query

12. Add the following columns to the query:
 - a. CLASS.ClassName
 - b. CLASS.Term
 - c. CLASS.Section
 - d. STUDENT.LastName
 - e. STUDENT.FirstName
 - f. GRADE.Grade
13. In the *Sort:* row of the Field properties, set the sort order for ClassNumber, Section, LastName, and FirstName to **Ascending**.
14. The complete query now appears as shown in Figure A-44.
15. Click the **Shutter Bar Open/Close** button to expand the Navigation Pane.
16. To save the query, click the **Save** button on the Quick Access Toolbar to display the **Save As** dialog box. Type in the query name **QBE-Query-A-01**, and then click the **OK** button. The query is saved, and the window is renamed with the new query name.
17. Click the **Run** button on the Query Design in the Result command group of the DESIGN command tab. The query results appear, as shown in Figure A-45.

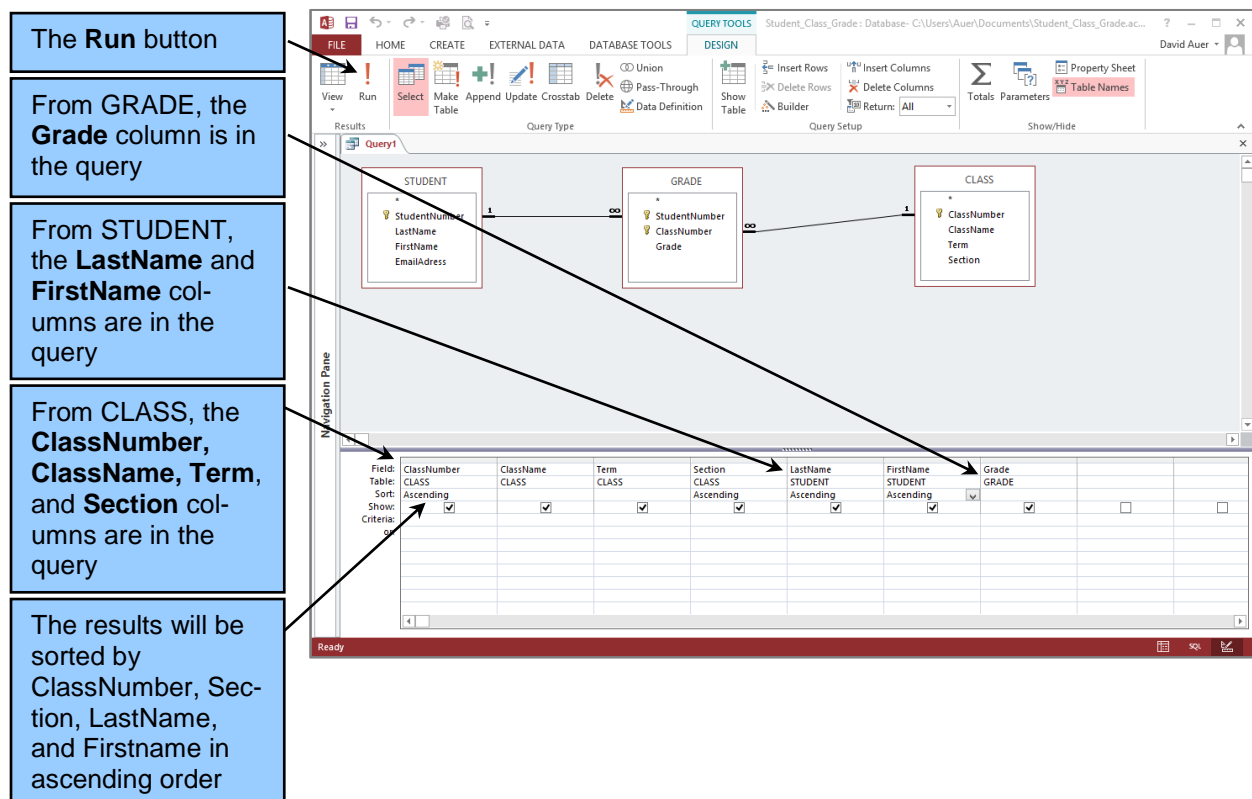


Figure A-44 — The Completed Three-Table QBE Query

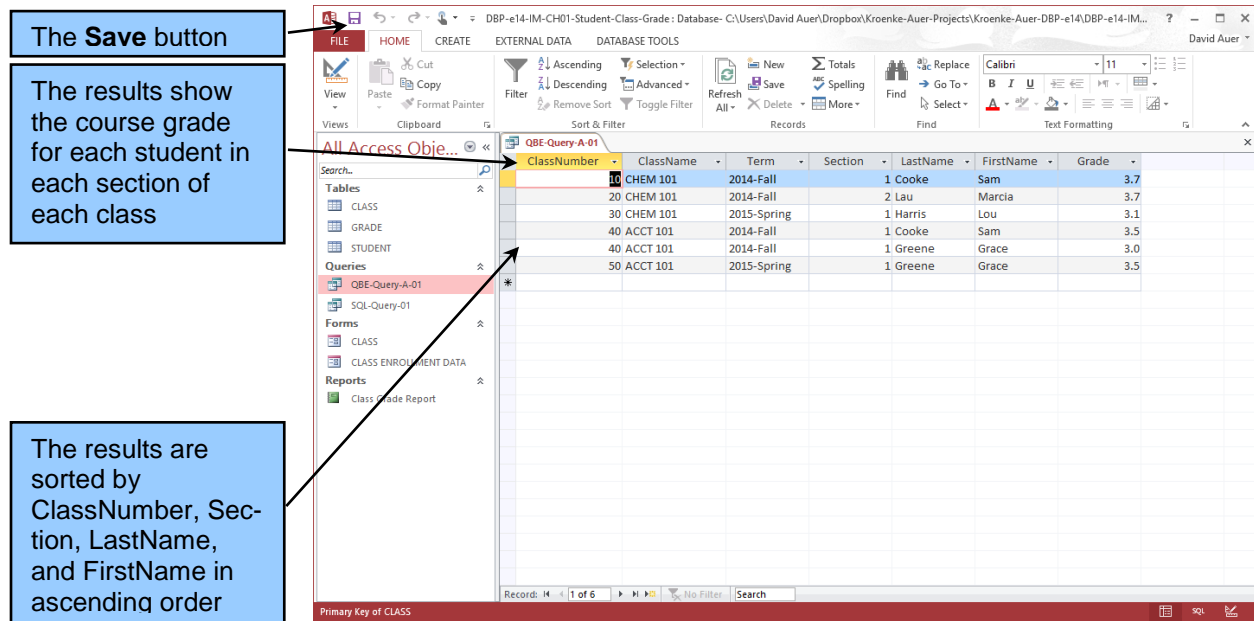


Figure A-45 — The Three-Table QBE Query Results

18. The query document window is now named *QBE-Query-A-01*, and a newly created QBE-Query-A-01 query object appears in a Queries section of the Navigation Pane.
19. Close the QBE-Query-A-01 query.
20. If Microsoft Access displays a dialog box asking whether you want to save changes to the design of the query QBE-Query-A-01, click the **Yes** button.
21. The Student-Class-Database now appears as shown in Figure A-46.

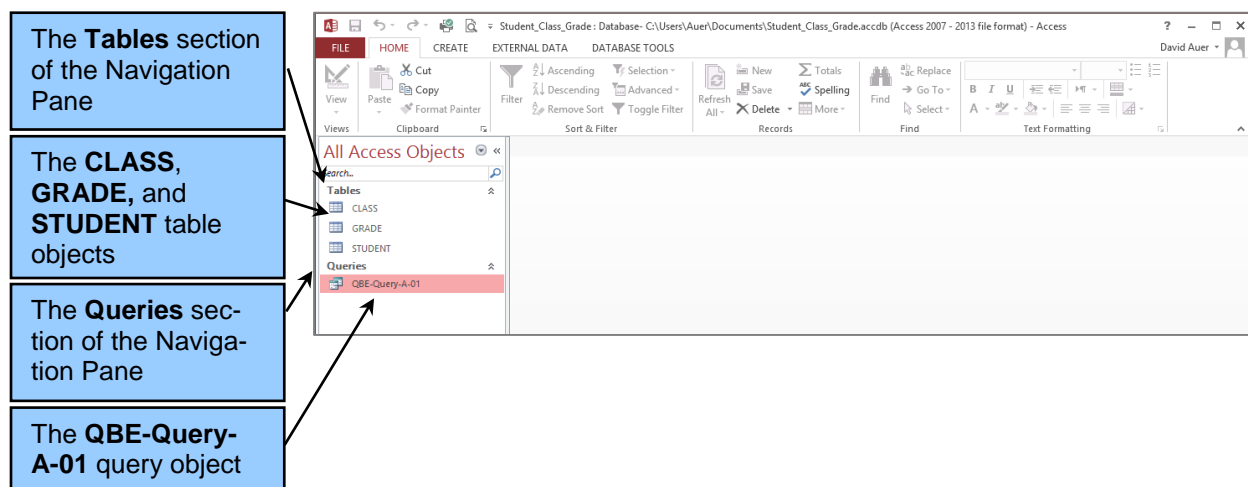


Figure A-46 — The Database with the QBE-Query-A-01 Query Object

This is a fairly complex query, but it certainly illustrates how to use QBE for more complicated queries. And if you can run this query, you will be able to run queries using just one or two tables.

How Do I Create Microsoft Access 2013 Forms and Reports?

Microsoft Access 2013 has the ability to create and store **forms** and **reports** as part of its application development tools. We will not discuss these features in detail, but we will note that Microsoft Access has both a **Form Wizard** and a **Report Wizard** that will step you through the creation of basic forms and reports. These provide a good starting place for exploring forms and reports in Microsoft Access 2013.

The Form Wizard button is located in the Forms command group on the CREATE ribbon, as shown in Figure A-47. Once started, the Form Wizard takes us through a step-by-step process to create the desired form.

The Report Wizard button is located in the Reports group on the Create ribbon, as shown in Figure A-48. Once started, the Report Wizard takes us through a step-by-step process similar to the one used by the Form Wizard, but this time to create the desired report.

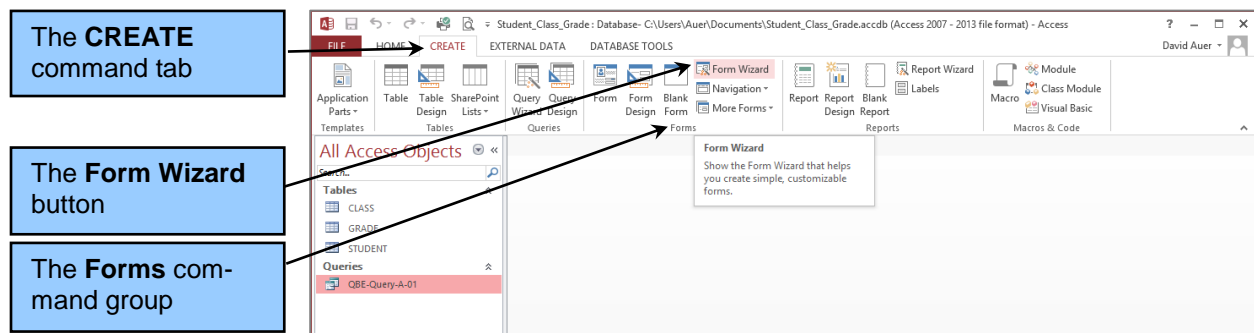


Figure A-47 — The Form Wizard

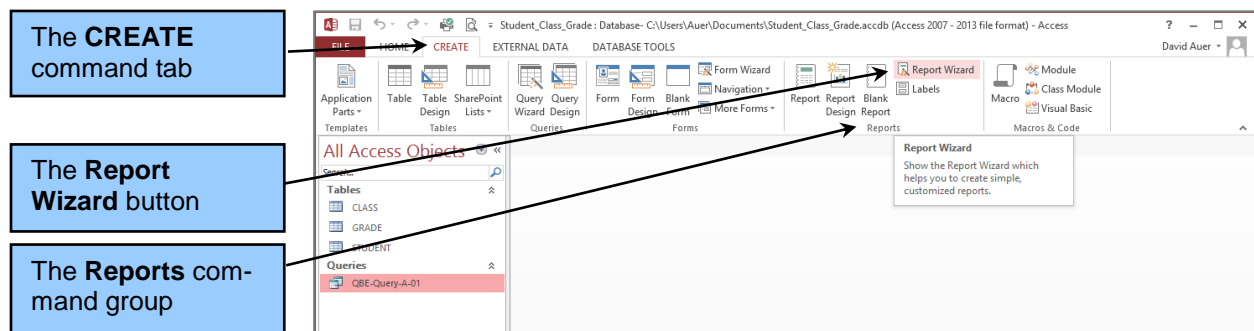


Figure A-48 — The Report Wizard

How Do I Close a Newly-Created Database and Exit Microsoft Access 2013?

We have finished all the work we need to do in this appendix on getting started with Microsoft Access 2013. We have demonstrated how to create a database, build database tables, populate a table with data by using Datasheet view, create relationships between tables, and query a database using Microsoft Access QBE. We have briefly introduced forms and reports.

At this point, you know enough to create and use basic Microsoft Access databases. In particular, you know enough to create the Cape Codd database that will be used in Chapter 2 in our discussion of SQL queries. In fact, creating that database is part of the exercises in this appendix.

Having achieved our goal of getting you started using Microsoft Access 2013, we finish by closing the Student-Class-Grade database and Microsoft Access 2013.

Closing the Student-Class-Grade Database and Exiting Microsoft Access 2013

1. To close the Student-Class-Grade database and exit Microsoft Access 2013, click the **Close** button in the upper-right corner of the Microsoft Access 2013 window.

Key Terms

AutoNumber	character
command tab	composite primary key
contextual command tabs	data entry form
datasheet	Datasheet view
Enforce Referential Integrity check box	foreign key
form	Form Wizard
key	Microsoft Office Fluent user interface
Navigation Pane	number
numeric	object
primary key	query
Query by Example (QBE)	Relationships window
remarks	report
Report Wizard	required
Ribbon	schema
surrogate key	text
Tools Relationships command	type

Review Questions

- A.1 Using the *Student-Class-Grade* database that you created in this Appendix:
- Create and run a Microsoft Access QBE query to duplicate the results in Figure 1-12. Save the query as QBE-Query-A-02.
 - Use the Form Wizard to create a data input form for the STUDENT table. Name the form *Student Data Input Form*. Using the student data shown in Figure A-49, add the new students to the STUDENT table.
 - Use the Form Wizard to create a data input form for the CLASS table. Name the form *Class Data Input Form*. Using the class data shown in Figure A-50, add the new classes to the CLASS table.
 - Use the Form Wizard to create a data input form for the GRADE table. Name the form *Grade Data Input Form*. Using the grade data shown in Figure A-51, add the new grades to the GRADE table.
 - Use the Form Wizard to duplicate the CLASS form in Figure 1-11. Note that this form uses more than one table.

StudentNumber	LastName	FirstName	EmailAddress
5	Davis	Bruce	Bruce.Davis@OurU.edu
6	Kelly	Mary	Mary.Kelly@OurU.edu
7	Taylor	Larry	Larry.Taylor@OurU.edu

Figure A-49 — Additional STUDENT Data

ClassNumber	ClassName	Term	Section
60	MATH 105	2014-Fall	1
70	MATH 105	2014-Fall	2
80	MATH 105	2014-Fall	3
90	MATH 110	2014-Spring	1

Figure A-50 — Additional CLASS Data

StudentNumber	ClassNumber	Grade
1	60	3.3
2	60	3.5
5	70	3.7
6	70	2.7
7	80	3.0
7	90	3.3

Figure A-51 — Additional GRADE Data

- F. Use the Report Wizard to create a report of the data in the STUDENT table. Name the report *Student Data Report*.
 - G. Use the Report Wizard and the CLASS, STUDENT, and GRADE tables to replicate the *Class Grade Report* shown in Figure 1-13. Note that your version of this report will display additional data because of the data you added to the database in steps B, C, and D above.
- A.2 In this exercise, you will build the Cape Codd database used for the SQL examples in Chapter 2. The Microsoft Access 2013 tables and relationships are shown in Figure 2-4.
- A. Create a new Microsoft Access 2013 database named *Cape_Codd.accdb*.
 - B. The column characteristics for the RETAIL_ORDER table are shown in Figure A-52. Using this data, create the RETAIL_ORDER table.
 - C. The column characteristics for the SKU_DATA table are shown in Figure A-53. Using this data, create the SKU_DATA table.
 - D. The column characteristics for the ORDER_ITEM table are shown in Figure A-54. Using this data, create the ORDER_ITEM table.
 - E. The data for the RETAIL_ORDER table is shown in Figure 2-6(a). Populate the RETAIL_ORDER table.
 - F. The data for the SKU_DATA table is shown in Figure 2-6(a). Populate the SKU_DATA table.
 - G. Create the relationship between the RETAIL_ORDER and ORDER_ITEM tables. Enforce referential integrity.
-

-
- H. Create the relationship between the SKU_DATA and ORDER_ITEM tables. Enforce referential integrity.
 - I. The data for the ORDER_ITEM table are shown in Figure 2-6(a). Populate the ORDER_ITEM table.
 - J. Why did you enter the ORDER_ITEM data only *after* creating the relationships between the tables?
 - K. Create a QBE query to display Department and Buyer from the SKU_DATA table. Save the query as *QBE-Query-A-RQ-01*.
 - L. The column characteristics for the CATALOG_SKU_20## table are shown in Figure A-55. Using this data, create the CATALOG_SKU_2014 and CATALOG_SKU_2015 tables.
 - M. The data for the CATALOG_SKU_2014 table is shown in Figure 2-6(b). Populate the CATALOG_SKU_2014 table.
 - N. The data for the CATALOG_SKU_2015 table is shown in Figure 2-6(b). Populate the CATALOG_SKU_2015 table.
 - O. Create a QBE query to display CatalogID, SKU, SKU_Description, and Department from the CATALOG_SKU_2014 table. Save the query as *QBE-Query-A-RQ-02*.
 - P. Create a QBE query to display CatalogID, SKU, SKU_Description, and Department from the CATALOG_SKU_2015 table. Save the query as *QBE-Query-A-RQ-03*.
-

RETAIL_ORDER

Column Name	Type	Key	Required	Remarks
OrderNumber	Number	Primary Key	Yes	Long Integer
StoreNumber	Number	No	No	Long Integer
StoreZip	Text (9)	No	No	
OrderMonth	Text (12)	No	Yes	
OrderYear	Number	No	Yes	Integer
OrderTotal	Currency	No	No	

Figure A-52 — Column Characteristics for the Cape Codd RETAIL_ORDER Table

SKU_DATA

Column Name	Type	Key	Required	Remarks
SKU	Number	Primary Key	Yes	Long Integer
SKU_Description	Text (35)	No	Yes	
Department	Text (30)	No	Yes	
Buyer	Text (30)	No	No	

Figure A-53 — Column Characteristics for the Cape Codd SKU_DATA Table**ORDER_ITEM**

Column Name	Type	Key	Required	Remarks
OrderNumber	Number	Primary Key, Foreign Key	Yes	Long Integer
SKU	Number	Primary Key, Foreign Key	Yes	Long Integer
Quantity	Number	No	Yes	Integer
Price	Currency	No	Yes	
ExtendedPrice	Currency	No	Yes	

Figure A-54 — Column Characteristics for the Cape Codd ORDER_ITEM Table

CATALOG_SKU_20##

Column Name	Type	Key	Required	Remarks
CatalogID	Number	Primary Key	Yes	Surrogate Key
SKU	Number	No	Yes	Integer
SKU_Description	Text (35)	No	Yes	
Department	Text (30)	No	Yes	
CatalogPage	Number	No	No	Integer
DateOnWebPage	Date	No	No	

Figure A-55 — Column Characteristics for the Cape Codd CATALOG_SKU_20## Table
