Microsoft® Office Access

Chapter 2: Using Design View, Data Validation, and Relationships

Access—Chapter 2 Using Design View, Data Validation, and Relationships

Chapter Overview

This chapter covers the basics of creating a table in *Design* view, using data validation rules to ensure your data is valid, and creating relationships among different tables in a database.

Student Learning Outcomes (SLOs)

After completing this chapter, students will be able to:

2.1. Create a table in *Design* view; add new fields; define a primary key; delete fields; save, close and open a table; and switch between *Datasheet* and *Design* views (p. A2-75).
2.2. Set field properties including *Field Size*, *Format*, *Caption*, *Default Value*, and *Required* (p. A2-79).
2.3. Explain data integrity and data validation concepts and options (p. A2-86).
2.4. Create field and record level validation rules, test rules, create validation text, create lookup fields, and define an input mask (p. A2-87).
2.5. Change field properties after records are added into a table (p. A2-99).
2.6. Explain relational database principles, understand the steps that must be taken to design a database, create relationships between tables, implement referential integrity, and delete and edit relationships (p. A2-104).
2.7. Import data records from Excel (p. A2-116).
2.8. Preview and print the contents of a table (p. 118).
2.9. Manage a database using *Compact & Repair* (p. A2-121).

Student Data Files

Project	Data File(s)	Solution File Name
Pause & Practice: Access 2-1	None	PP A2-1.accdb
Pause & Practice: Access 2-2	PP A2-1.accdb	PP A2-2.accdb
Pause & Practice: Access 2-3	PP A2-2.accdb	PP A2-3.accdb

Pause & Practice: Access 2-4	PP A2-3.accdb ServiceHistoryDataFile-02.xlsx	PP A2-4.accdb
Guided Project 2-1	CentralSierra-02.accdb	Access 2-1.accdb
Guided Project 2-2	SanDiegoSailing-02.accdb SDRentalsDataFile-02.xlsx	Access 2-2.accdb
Guided Project 2-3	AmericanRiver-02.accdb RaceResultsDataFile-02.xlsx	Access 2-3.accdb
Independent Project 2-4	CourtyardMedicalPlaza-02.accdb RentInvoicesDataFile-02.xlsx	Access 2-4.accdb
Independent Project 2-5	LifesAnimalShelter-02.accdb	Access 2-5.accdb
Independent Project 2-6	NewYorkDMV-02.accdb	Access 2-6.accdb
Improve It Project 2-7	PlacerHills-02.accdb	Access 2-7.accdb
Challenge Project 2-8	[your initials] Access 1-8.accdb	None
Challenge Project 2-9	None	None
Challenge Project 2-10	[your initials] Access 1-10.accdb	None

Chapter Outline

Case Study

Mary's Rentals is a privately owned equipment rental company. For the Pause & Practice projects in this chapter, students develop a database that contains information about the company's products. Students will use *Table Design* view to add new fields, define the primary key, edit field properties, and define validation rules, lookup fields and input masks. Students will define relationships and import data.

SLO 2.1—Creating a Table in Design View

(PowerPoint slides 4–5)

Design view is used to create a table from scratch. In this section, students will create a table in *Design* view, add new fields, define a primary key, delete fields, and save a table.

Create a New Table in Design View

You can add a new table into a database from the *Create* tab. A table created using the *Table Design* button automatically opens in *Design* view. The *Design* view window is divided into two areas. The top half displays all the fields in the table, including *Field Name*, *Data Type*, and *Description*. The lower half displays the detailed field properties of the selected field.

Add New Fields

Add new fields to a table by entering a field name and data type in the first empty row in the field list.

Define a Primary Key

A ***primary key*** is a field that contains a unique value for each record. When a table is created in *Design* view, a primary key field is not automatically assigned. You may use an existing field or create a field with the *AutoNumber* data type. To define the field as a primary key, use the *Primary Key* button in the *Tools* group of the *Table Tools Design* tab.

Delete Fields in a Table

If you have queries, forms, and reports that contain a field that you delete, you must also modify those objects to remove the deleted field.

Save, Close, and Open a Table

The save, close, and open operations on a table work the same whether you are in *Datasheet* or *Design* view.

Switch between Datasheet and Design Views

To switch between *Datasheet* and *Design* views use the *View* button on the *Ribbon* or the view icon located on the right side of the *Status* bar.

Teaching Tips and Suggestions

- Explain to students that the data type determines what field values can be entered in a field.
- Discuss the importance of the primary key field. If an existing field is not suitable for a primary key, an *AutoNumber* field may be used. When the primary key from one table is included as a field in a second table, the tables can be joined to create a relationship.
- Warn students that when a field is deleted all the field values are deleted. If there are queries, forms, or reports that contain the deleted field, these objects must also be modified to remove the field.
- Shortcut: To open the *Save As* dialog box, use **Ctrl+S**, the *Save* icon on the *Quick Access* toolbar, or the **F12** function key.

SLO 2.2—Setting Field Properties

(PowerPoint slides 6–11)

Field properties define the characteristics of each field. The available properties vary based on the data type of the field. This section introduces *Field Size*, *Format*, *Caption*, *Default Value,* and Required. These properties, if available for a field's data type, are located on the *General* tab in the lower half of the screen.

Set the Field Size

The *Field Size* property determines the maximum number of characters allowed in a *Short Text* field or the range of possible values in a *Number* field. A *Short Text* field can have a size from 0 to 255. A *Number* field can have seven possible field sizes that are explained in the following table:

Number Data Type Field Sizes

Unit	Description
Byte	Stores numbers from 0 to 255. Does not use decimal places.
Integer	Stores numbers from 232,768 to 32,767. Does not use decimal places.
Long Integer	Stores numbers from 22,147,483,648 to 2,147,483,647. Does not use decimal places. This is the default field size for a number data type.
Single	Stores numbers from 23.4 3 1038 to 3.4 3 1038 with a precision of 7 significant digits.
Double	Stores numbers from 21.797 3 1038 to 1.797 3 1038 with a precision of 15 significant digits.
Replication ID	Stores a globally unique identifier (FUID) randomly generated by Access.
Decimal	Stores numbers from 21028 to 1028 with a precision of 28 significant digits.

AutoNumber fields can use either *Long Integer* or *Replication ID*. The other number data types do not have a field size property.

The Format Property

The ***Format property*** indicates the way a field displays in a table.

Number, *Currency*, *Date/Time*, and *Yes/No* fields have predefined formats from which you can select. *Short Text* and *Long Text* fields do not have any predefined formats. You can use formatting symbols to create a custom format and change the way the data display. Common formatting symbols for *Short Text* and *Long Text* are described in the following table:

Short Text and Long Text Field Formatting Symbols

Character	Description
<	Characters display in lowercase. This must be the first of the formatting symbols if multiple symbols are used.
>	Characters display in uppercase. This must be the first of the formatting symbols if multiple symbols are used.
@	Enter one @ symbol for each character in the size of the field. If the field content contains fewer characters than the field size, the content displays with leading spaces.
[color]	Applies a color to the field. The color must be enclosed in brackets. The possible colors are black, blue, cyan, green, magenta, red, yellow or white. This formatting symbol only works when applied in conjunction with one of the other formatting symbols.

The Caption Property

The ***Caption property*** contains the text that displays in the column header to identify a field in *Datasheet* view of tables, forms, queries, and reports. Captions often include spaces and multiple words as compared to the field name. Access displays the field name as the caption if you do not enter a caption value.

Set the Default Value

You can use the *Default Value* property to automatically insert a specified value into a new record in a table.

Set Required Field

You can specify whether a user must enter a value into a field or whether he or she can skip the field and leave it blank. If a field must have a value, you should set the *Required* property to *Yes*. The default value for the *Required* property is *No* except for the primary key field which is always required.

PAUSE & PRACTICE: ACCESS 2-1

File Needed: None
Completed Project File Name: ***[your initials] PP A2-1.accdb***

For this project students will create a database for Ryan Thomas, operations director for Mary's Rentals. Students will use *Design* view to create a table to store information about the rental equipment. In addition, students will modify the *Field Properties* to enhance the functionality of this table.

Teaching Tips and Suggestions

- Recommend that students choose a *Short Text* data type for fields not used in arithmetic operations, even if the field contains numbers.
- Explain to students that they should set the field size property based on the largest value expected. Access processes smaller data sizes faster, using less memory, optimizing performance and storage space.

- Note that *Short Text* and *Long Text* data types default to display left-justified. The @ formatting symbol does not change the alignment, but when leading spaces are included for field contents that are less than the field size, the displayed content may not appear to be left-justified.
- Discuss why students should use the *Caption* property instead of putting spaces in field names. Although Access allows spaces in field names, it is not recommended. Spaces in field names cause problems when you attempt to perform more advanced tasks in manipulating fields.
- Mention that the *Default* value property only affects new records not existing records.
- Explain to students that while building tables, it is recommended to set the *Required* property last. This allows you to easily test different features in the table without being forced to enter a value for every field. Note that data entry is always required in the primary key field.
- Shortcut: Press **Ctrl+Enter** to force a line break in a text field.

SLO 2.3—Understanding Data Integrity and Data Validation

(PowerPoint slides 12–13)

The process of verifying the accuracy, or integrity, of the data is known as ***data validation***. Data validation uses the data integrity rules that have been defined in the field and table properties. Data integrity rules ensure that the data in a database is accurate and consistent. Data integrity rules are included as part of the metadata, the descriptions about what the different data fields represent and their formats. In Access, metadata is entered into the different field properties. Common data integrity rules are described in the following table.

Data Integrity Rules

Rule Focus	**Description**
Data Format	Ensures that a value entered into a field matches the data type established for that field; meets any size constraints; and meets any required formatting, such as beginning with a number or letter, or matching a month/day/year format.
Range	Ensures that a value entered into a field falls within the range of acceptable values that have been established.
Consistency	Ensures that a value entered into a field is consistent with a value in another field.
Completeness	Ensures that a field contains a value if it is required.

Teaching Tips and Suggestions

- Show students an example of validation rules and validation text in the “Order Details” table in Northwind Traders, as shown in Figure 2-21. Northwind Traders is a sample database that comes with Access.

SLO 2.4—Integrating Data Validation in Tables

(PowerPoint slides 14–20)

Data Format, *Range*, and *Consistency* data integrity rules can be created using field and record validation rules. Validation rules can be created in *Design* view or *Datasheet* view. Lookup fields ensure range integrity and *Input Masks* help users enter data in the correct format.

Field Validation Rules

Field validation rules limit entries to a certain range of values or format. Access checks field validation rules when a user navigates out of the field. If the data violates the rule, Access requires the user to fix the problem before he or she can leave the field. Operators used in validations rules, or expressions, are listed in the following table:

Operators Used in Access Expressions

Arithmetic	Comparison	Logical
-	<	AND
+	<=	OR
*	>	NOT
/	>=	XOR
^	=	
	<>	
	Between	
	In	
	Like	

Use the *Like* comparison operator to find a value that is similar to (or like) something else. When using *Like*, you typically incorporate one of the wildcard characters available in Access. When using wildcards, the criteria of the expression must be enclosed in quotation marks.

Using Wildcard Character in Validation Rules

Wildcard Character	Matches	Example	Explanation of Example
?	Any single character (numbers are also considered characters)	Like “C????”	Values entered must be 5 characters long and begin with the letter C. *(Tip: Even if the field size is larger than 5, this rule requires that exactly 5 characters are entered.)*
*	Any number of characters	Like “MA*”	Values entered must begin with MA, but can be any length up to the maximum field size.
#	Any single numeric digit	Like “##”	Values entered must contain two numeric digits.

Record Validation Rules

Record validation rules compare fields in the same table. They can also be used to create range and data format integrity rules by limiting entries to a certain range of values or a specified format. Record validation rules are entered into the *Validation Rule* property of the table. The rule is checked when a user attempts to navigate out of the record. If the data the user has entered in the record violates the rule, Access requires the user to fix the problem before he or she can leave the record.

Examples of entries for record validation rules, along with an explanation of the effects, are shown in the following table:

Using and Understanding Record Validation Rules

Record Validation Rule Example	Explanation of Effect
([Field1] is Null) XOR ([Field2] is Null)	You must enter a value into either *Field1* or *Field2*, but not into both fields. *(Tip: The* XOR *logical operator prevents both fields from having a value; both fields would be allowed to have a value if OR were used instead.)*
[Field1] <> [Field2]	You may not enter the same value into both *Field1* and *Field2*.
(([Field1] > [Field2]) AND [Field2]>10))	*Field1* must be greater than *Field2* and *Field2* must have a value greater than 10.

Test Validation Rules

When using validation rules it is important that the rules are tested. These tests ensure that the logic is valid and the rule works as you intended.

Test Data Examples for Validation Rules

Validation Rule	Examples of Test Data	Result of Test
“M” OR “F”	M F X	Valid (test of M condition) Valid (test of F condition) Invalid
>=#1/1/2015# AND <#1/1/2016#	12/15/2013 7/7/2015 2/13/2016	Invalid (below the range) Valid Invalid (above the range)
([Field1] is Null) XOR ([Field2] is Null)	*Field1* null and *Field2* null *Field1* = 5 and *Field2* = 7 *Field1* null and *Field2* = 7 *Field1* = 5 and *Field2* null	Invalid (both can’t be null) Invalid (both can’t have a value) Valid (only *Field2* has a value) Valid (only *Field1* has a value)

Create Validation Text

When data is entered that violates a validation rule, Access automatically displays a message box alerting the user to the problem. Use the Validation Text property to enter a custom message that displays instead of the default message.

Create a Lookup Field

Lookup fields display a list of data values from which the user can choose. Lookup fields are another way to implement range integrity rules and are an alternative to validation rules since they also limit what can be entered into a field. Lookup fields can be used on *Number*, *Short Text*, and *Yes/No* fields.

Text and *Number* fields have three options for the *Display Control* property. The way the control functions varies, depending on whether you are viewing the data in a table or a form.

Understanding Display Control Options

Display Control Choice	Explanation
Text Box	Displays the contents of a field and also allows the user to type a value into that field.
List Box	Table view: Displays as a drop-down list. Form view: Displays as a box showing all possible choices.
Combo Box	Displays as a drop-down list in both table view and form view.

Define an Input Mask

An ***input mask*** forces a user to enter data using a specific format. An input mask can only be used on fields with data types of *Short Text, Number* (excluding *ReplicationID*), *Currency,* or *Date/Time.*

Input Mask Special Characters

Character	Explanation of Effect
0	User must enter a number (0 to 9).
9	User can enter a number (0 to 9), but it is not required.
#	User can enter a number (0 to 9), space, or plus or minus sign, but it is not required. If not filled in, Access enters a blank space.
L	User must enter a letter (A to Z, upper or lower case).
?	User can enter a letter (A to Z, upper or lower case), but it is not required.
A	User must enter a letter or a number.
a	User can enter a letter or a number, but it is not required.
!	In a text field, when fewer characters are entered than specified in the mask, the data displays right-justified with empty spaces on the left side of the mask instead of the default right side of the mask.
“”	Characters entered inside of the quotation marks display as written.
\	Characters following the slash are displayed as written.

An input mask contains three parts, each separated by a semicolon.

- The first part is mandatory and shows the desired formatting.
- The second part of the input mask is optional and indicates whether the mask characters, such as parentheses, dashes, etc., are stored with the data. Not storing the characters saves space; this can be significant in large databases. A 0 tells Access to store the characters; a 1 or empty tells Access not to store the characters.
- The third part of the mask is also optional and specifies what symbol displays in the mask as the user enters the data. The default symbol is the _ (underscore).

Teaching Tips and Suggestions

- Inform students that validation rules cannot be used on fields with data types of *Attachment*, *AutoNumber*, *OLE Object,* and the *ReplicationID Field Size* choice of a *Number* field.
- If students want a validation rule to compare two or more fields, explain that this is accomplished through a record validation rule.

- Note that validation rules ignore the case of the text.
- Explain how the *Yes/No* data type stores values. Even though the *Yes/No* data type displays data using a check box, Access stores Yes as -1 and No as 0. Expressions written to check the contents of a *Yes/No* data type should compare the contents to either -1 or 0.
- Show students the expression builder dialog box by clicking the *Build* button in the *Validation Rule* property.
- Show an example to help students understand the difference between *XOR* and *OR*. The *XOR* logical operator prevents both fields from having a value; both fields would be allowed to have a value if *OR* were used instead.
- Encourage students to test each validation rule as it is written to avoid debugging more than one at a time.
- If students create a validation rule after data is already in the table, recommend that they test to ensure that the data already in the table is verified against the new rules. Show students the *Test Validation Rules* button.
- Recommend that students use combo boxes in forms instead of list boxes. List boxes take up too much room and are not intuitive.
- Ask students for examples of field data that could use an input mask.
- Shortcut: In the *Validation Rule* property, **Ctrl+F2** opens the *Expression Builder*.
- Shortcut: In the *Validation Text* property, **Shift+F2** opens the *Zoom* window.

SLO 2.5— Changing Field Properties after Records Are Added into a Table

(PowerPoint slides 21–22)

Making changes to the table design after records are added must be done with caution. Examples of some of the potential problems are described below.

If you reduce the *Field Size*, Access warns you that data may be lost. If you click *Yes* to continue, Access truncates the contents of any data values that are larger than the new field size. Be certain that the field is large enough so that you don't lose any data since you can't undo this change.

If you change the *Required* property to *Yes*, Access warns that data integrity rules have changed. If you click *Yes* to test the data with the new rules, Access validates the existing data. This could take a long time depending on the number of records in the table. If any records violate the new rule, another message box displays. You must decide whether to keep the new rule. This message only appears once to let you know that there is data that violates the rule, not once for each record that violates the rule. If you proceed with the change, you must open the table and individually correct each violation of this new rule.

Pause & Practice: Access 2-2

File Needed: ***[your initials] PP A2-1.accdb***
Completed Project File Name: ***[your initials] PP A2-2.accdb***

For this project, students add features to enhance the data integrity in the *Equipment* table. Students add field and record level validation rules and validation text, use the *Lookup Wizard* to create a combo box, and add an input mask. Students will test the validation rules and add data into the *Equipment* table.

Teaching Tips and Suggestions

- In the Pause & Practice steps 6a and 8a, explain to students that they don't need to test the data integrity rules because we know that the existing data complies with the new rules.
- Shortcut: In any of the field properties that have a drop-down list, double-click to cycle through the choices. For example, in the *Required* field property double-click to toggle from *Yes* to *No*.

SLO 2.6— Understanding and Designing Relational Databases

(PowerPoint slides 23 –27)

In this section, students will learn how to create a collection of integrated and related tables.

Relational Principles

The data in a relational database is organized into a collection of related tables. The tables are related, or connected, to each other through common fields. You can create these relationships and view them in the *Relationships* window. To be considered relational, a database must meet certain specifications.

- The data in the database must be stored in tables.
- The rows of the table must represent records.
- The columns of the table must represent fields.
- Each row must have a unique identifier or primary key. The primary key allows the DBMS to locate a specific record in the database. The primary key can either be one field in the table or a combination of fields.
- Each table must have a relationship with at least one other table. In order to have a relationship, the two tables must have a field in common with each other. This common field is created by adding the primary key field from one table into the related table. This added field is known as a ***foreign key***. You create a relationship between the two tables by linking the foreign key to the primary key.

The Steps to Design a Relational Database

1. Determine the purpose of the database and the scope of the functions that will be performed.
2. Determine the major objects that are needed in the database to support its purpose and functions.
3. Determine the specific details you need to capture about each object.
4. Determine which field in each table is the primary key.
5. Determine the data type and size of each field.
6. Determine any additional restrictions on what kinds of data can be stored in the fields.
7. Determine how the different tables are related to each other.

Three Types of Relationships

There are three different types of relationships that may exist between tables in a database: one-to-one, one-to-many, and many-to-many. One-to-many relationships are the most common.

In a ***one-to-one relationship***, one row of data in *Table A* may be associated with only one row of data in *Table B*. See Figure 2-56 for an example.

In a ***one-to-many relationship***, one row of data in *Table A* may be associated with many rows of data in *Table B*. This type of relationship is most common. In the relationships window the infinity symbol is used to represent the many side of the relationship. See Figure 2-57 for an example.

In a ***many-to-many relationship***, many rows of data in *Table A* may be associated with many rows of data in *Table B*. A relational database does not directly support M:N relationships. They require the creation of a third table, *Table C*. This third table is a junction or intersection table that matches up the records across the two tables. See Figure 2-58 for an example.

In order to create these relationships, there must be a field that is common between the two tables. This field would be the primary key in one table and the foreign key in the related table.

For a 1:M relationship, you create the foreign key by taking the primary key field from the 1 table and adding it as an extra field into the M table. This added field is known as the foreign key.

For a 1:1 relationship, you create the foreign key by taking the primary key from either table and adding it into the other table.

If you determine that you have a M:N relationship, you first need to create the junction table and convert the M:N relationship into two 1:M relationships. The ***junction table*** contains the primary key fields from each of the 1 tables, along with any other fields that describe that junction.

When adding the foreign key field to the related table, set the field properties to be the same *Data Type* and *Field Size* as the primary key in the 1 table. The foreign key field name can be anything. Often it will be either the exact same name as the primary key or the name of the primary key preceded by FK.

Create Relationships between Tables

After determining and defining the primary key and foreign key fields, the relationships may be created. This should be completed before adding any data into the tables that contain the foreign keys. Create the relationship in the *Relationships* window, accessible from the *Relationships* button on the *Database* tools tab. The fields are related by dragging the primary key field on top of the foreign key field in the related table.

When creating a relationship, you must decide if you want to enforce referential integrity. ***Referential integrity*** ensures that records in associated tables have consistent data. This ensures consistency and integrity of the database. If you do not enforce referential integrity, you can have records in the associated table that do not match a record in the 1 table.

If referential integrity has been enforced, Access prohibits two other actions:

- You cannot delete a record from the 1 table if that record has related records in the related table.
- You cannot change the value of the primary key field in the 1 table if that record has related records in an associated table.

The only way to override these constraints is to choose *Cascade Update Related Fields* and *Cascade Delete Related Records* when defining the relationship.

Delete a Relationship between Tables

To remove a relationship, close all of the tables involved in the relationship, as well as any objects that are based on those tables. Go into to the *Relationships* window, right-click on the relationship line, and choose **Delete**.

Edit a Relationship between Tables

Go into to the *Relationships* window, right-click on the relationship line, and choose **Edit Relationship**.

PAUSE & PRACTICE: ACCESS 2-3

File Needed: ***[your initials] PP A2-2.accdb***
Completed Project File Name: ***[your initials] PP A2-3.accdb***

For this project, students add a second table to store the service history of the equipment and define a relationship between the *Equipment* and *Service History* tables.

Teaching Tips and Suggestions

- Step 1, in "The Steps to Design a Relational Database," should be used to ensure that data outside of the scope of the database is not included.
- Step 2, in "The Steps to Design a Relational Database," minimizes the amount of duplicate or redundant data.
- While discussing step 4, in "The Steps to Design a Relational Database," explain composite keys. For some tables two fields are needed to uniquely identity each row and function as the primary key. If a primary key is made up of more than one field it is called a ***concatenated key*** or ***composite key***.
- Steps 5 and 6, in "The Steps to Design a Relational Database," help to increase data integrity. Give examples of data integrity such as limiting field size, setting the required property, carefully choosing the data type, and limiting values that can be entered.
- Show students the shorthand notation for describing relationships.
    - 1:1 (one-to-one)
    - 1:M (one-to-many)
    - M:N (many-to-many)
- In a many-to-many relationship, the third table, called the junction or intersection table, may also be referred to as a composite, bridge, or associative table.
- When explaining relationships and foreign keys, show students the difference between Figure 2-59 and Figure 2-60. The tables shown in Figure 2-59 can be found in the Life's Animal Shelter database in the Chapter 1 data files.
- Inform students that when creating a 1:1 relationship you must set the *Indexed* property of the foreign key field to *Yes (No Duplicates)* for Access to recognize a 1:1 relationship.

- Explain to students that the foreign key field must have the same data type and field size as the primary key in the related table. The field name does not have to be the same.
- Explain to students that Access automatically identifies the relationship type as 1:1 or 1:M. Show students as it is displayed in the *Edit Relationships* dialog box.
- Independent Project 2-6 is a good example illustrating the need to enforce referential integrity.

SLO 2.7— Importing Data Records from Excel

(PowerPoint slides 28-29)

In this section, students learn how to import data from an Excel file into a table that they have already created. Prior to importing the file, they need to ensure that the Excel file is formatted correctly.

- The column headings in the first row of the Excel file must match existing field names in the Access table.
- The data fields don't need to be in the same order.
- All of the fields in the table do not need to be in the Excel file as long as the *Required* property allows the field to be empty.
- If the fields are not of the same data type (for example the Access field data type is *Number* but the Excel file field contains text), Access still imports the file but the contents of that field is left empty. In this case, the *Required* property must allow the field to be empty.

Teaching Tips and Suggestions

- In this section, students learn how to import data from an Excel file into a table that has already been created. Explain to students that they can also import data into a new table or simply link to the Excel file.
- When students select the *Append* option while selecting the source and destination of the data, remind them to select the destination table in the drop-down list.

SLO 2.8—Printing the Contents of a Table

(PowerPoint slides 30–31)

Table data may be printed using the *Print* button in *Backstage* view. There are three print options:

- ***Quick Print:*** Sends the contents of the current object immediately to the printer.
- ***Print:*** Allows you to select different print options before sending the contents to the printer.
- ***Print Preview:*** Allows you to preview the way the table records will print before actually sending them to the printer.

Preview the Data Records

Previewing the way that your table records will print is a good idea. This allows you to check that the formatting is appropriate and to make changes to any settings before printing.

Print the Data Records without Previewing

If you previously previewed how the table will print, you can quickly print the table records without previewing the pages.

SLO 2.9—Managing a Database Using the Compact & Repair Database Utility

(PowerPoint slides 32–33)

Use the *Compact & Repair Database* utility to reduce the chances of corruption and to reclaim unused space from temporary and deleted objects. If this is a multi-user database, no one can be using the database at the time the *Compact & Repair* is performed. Access the *Compact & Repair* option from the *File* tab.

Pause & Practice: Access 2-4

Files Needed: ***[your initials] PP A2-3.accdb* and *ServiceHistoryDataFile-02.xlsx***
Completed Project File Name: ***[your initials] PP A2-4.accdb***

For this project, students import data into the *Service History* table, print the contents of the *Equipment* table and manage the database using the *Compact & Repair Database* option.

Guided Project 2-1

File Needed: ***CentralSierra-02.accdb***
Completed Project File Name: ***[your initials] Access 2-1.accdb***

For this project, students enhance the Central Sierra Insurance database. Students will use *Design* view to create a second table, edit field properties, and integrate data integrity rules. Students will create a relationship between the two tables and enforce referential integrity constraints. Finally, students will view how the table records will print using print preview. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8]

Guided Project 2-2

Files Needed: ***SanDiegoSailing-02.accdb* and *SDRentalsDataFile-02.xlsx***
Completed Project File Name: ***[your initials] Access 2-2.accdb***

For this project, students enhance the San Diego Sailing Club database. Students will use *Design* view to create a second table, edit field properties, and integrate data integrity rules. Students will also create a relationship between the two tables, enforce referential integrity constraints, and import data into the second table. Finally, students *Compact & Repair* the database. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9]

GUIDED PROJECT 2-3

Files Needed: ***AmericanRiver-02.accdb*** **and** ***RaceResultsDataFile-02.xlsx***
Completed Project File Name: ***[your initials] Access 2-3.accdb***

For this project, students enhance the American River Cycling Club database. Students will use *Design* view to create a second table, edit field properties, and integrate data integrity rules. Students also create a relationship between the two tables and enforce referential integrity constraints. Finally, students import data and view how the table records will print using print preview. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6, 2.7]

Teaching Tips and Suggestions

- Review with students the need for two fields for the primary key, making it a concatenated, or composite, primary key.

INDEPENDENT PROJECT 2-4

Files Needed: ***CourtyardMedicalPlaza-02.accdb*** **and** ***RentInvoicesDataFile-02.xlsx***
Completed Project File Name: ***[your initials] Access 2-4.accdb***

For this project, students enhance the Courtyard Medical Plaza database. Students will use *Design* view to create a second table, edit field properties, integrate data integrity rules, and enter data. Students also create a relationship between the two tables, enforce referential integrity constraints, and import data. Finally, students preview how the records will print. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6, 2.7]

INDEPENDENT PROJECT 2-5

File Needed: ***LifesAnimalShelter-02.accdb***
Completed Project File Name: ***[your initials] Access 2-5.accdb***

For this project, students enhance the Life's Animal Shelter database. Students will use *Design* view to create a second table, edit field properties, integrate data integrity rules, and enter data. Students also create a relationship between the two tables and enforce referential integrity constraints. Finally, students add data using the relationship between tables. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6]

INDEPENDENT PROJECT 2-6

File Needed: ***NewYorkDMV-02.accdb***
Completed Project File Name: ***[your initials] Access 2-6.accdb***

For this project, students enhance the New York Department of Motor Vehicles database. Students will use *Design* view to edit a table and integrate data integrity rules. Students also create a relationship between the two tables and enforce referential integrity constraints. Finally, students view how the table records will print using print preview. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8]

Teaching Tips and Suggestions

- This project is a good example illustrating the need to enforce referential integrity.

Improve It Project 2-7

File Needed: ***PlacerHills-02.accdb***
Completed Project File Name: ***[your initials] Access 2-7.accdb***

For this project, students enhance the Placer Hills Real Estate database. Students will use *Design* view to edit properties in a table and integrate data integrity rules. Students edit data records, create a relationship between the two tables, and enforce referential integrity constraints. Finally, students view how the table records will print using print preview. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6, 2.8]

Teaching Tips and Suggestions

- In step 6, students are determining what needs to be done to relate the two tables. Students are not actually making any changes to define the relationship until step 7.
- Discuss with students that it is good practice to define the relationships before adding data to a database, if possible.

Challenge Project 2-8

File Needed: ***[your initials] Access 1-8.accdb***
Completed Project File Name: ***[your initials] Access 2-8.accdb***

For this project, students will modify the database created in Challenge Project 1-8. Students will use *Design* view to edit field properties. Students will change any data that violates these properties, preview the table for printing, and *Compact & Repair* the database. [Student Learning Outcomes 2.2, 2.3, 2.4, 2.5, 2.8, 2.9]

Challenge Project 2-9

File Needed: None
Completed Project File Name: ***[your initials] Access 2-9.accdb***

For this project, students will create a database for the Blue Lake Sports sporting goods company's internship program. Students will create two tables, create a relationship between the tables, and add data into the database. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6]

Challenge Project 2-10

File Needed: ***[your initials] Access 1-10.accdb***
Completed Project File Name: ***[your initials] Access 2-10.accdb***

For this project, students will modify the database created in Challenge Project 1-10 to be used for a travel agency. The database will contain a table about upcoming cruises. Students will use *Design* view to create a new table to store information about the different ships on which passengers can cruise. Students will create relationships between the tables, and add data into the new table. [Student Learning Outcomes 2.1, 2.2, 2.3, 2.4, 2.6]