Objectives

• Session 5.1
  – Review object naming standards
  – Use the Like, In, Not, and & operators in queries
  – Filter data using an AutoFilter
  – Use the IIf function to assign a conditional value to a calculated field in a query
  – Create a parameter query
Objectives (Cont.)

• Session 5.2
  – Use query wizards to create a crosstab query, a find duplicates query, and a find unmatched query
  – Create a top values query
Objectives (Cont.)

- Session 5.3
  - Modify table designs using lookup fields, input masks, and data validation rules
  - Identify object dependencies
  - Review a Long Text field’s properties
  - Designate a trusted folder
Creating Advanced Queries and Enhancing Table Design

- **Case - Chatham Community Health Services**
  
  *Making the Clinic Database Easier to Use*

- User is interested in taking better advantage of the power of Access to make the database easier to use and to create more sophisticated queries

- User also needs a summarized list of invoice amounts by city
Creating Advanced Queries and Enhancing Table Design (Cont.)

STARTING DATA FILES

Access2
- Tutorial
  - Clinic.accdb
- Case2
  - Tutoring.accdb

Review
- Supplier.accdb
- Case3
  - Rosemary.accdb

Case1
- Task.accdb
- Case4
  - Ecotour.accdb
Creating Advanced Queries and Enhancing Table Design (Cont.)

The tbl prefix tag identifies a table object.

The qry prefix tag identifies a query object.

The frm prefix tag identifies a form object.

The rpt prefix tag identifies a report object.

A Select query selects the records in the fields that satisfy the criteria.

A calculated field contains an expression that calculates the values of the data in the field.

The design grid contains the fields and criteria that will be used in the query.

The name of the new calculated field is placed to the left of the expression, separated with a colon.

The If function tests a condition and returns one of two values. The function returns the first value if the condition is true, and the second value if the condition is false.

The Expression Builder can be used to create an expression for a calculated field.

The false part of this IF function will concatenate the data in the Parent field to the string "(Parent)"

The true part of this IF function will concatenate the data in the LastName field to a string with a comma and space. It will then concatenate the result to the data in the FirstName field.

The IsNull function will return a value of true if the field is empty and will return a value of false if the field is not empty.
Reviewing the Clinic Database

- The Navigation Pane displays the objects grouped by object type
  - Each object name has a prefix tag—a tbl prefix tag for tables, a qry prefix tag for queries, a frm prefix tag for forms, and a rpt prefix tag for reports
  - All three characters in each prefix tag are lower case. The word immediately after the three-character prefix begins with an upper case letter
  - Using object prefix tags, you can readily identify the object type, even when the objects have the same base name
  - Object names have no spaces, because other database management systems do not permit making it easy during conversions to those systems
Using Pattern Match in a Query

• A **pattern match** selects records with a value for the designated field that matches the pattern of a simple condition value

• The **Like comparison operator** selects records by matching field values to a specific pattern that includes one or more of these wildcard characters: asterisk (*), question mark (?), and number symbol (#)
  
  – The asterisk represents any string of characters, the question mark represents any single character, and the number symbol represents any single digit
Using Pattern Match in a Query (Cont.)

**Figure 5-2** Record selection based on matching a specific pattern

Pattern match selection criterion

**Figure 5-3** tblPatient table records for area code 860

Scroll down to see more records that match the criteria

46 records total
Using a List-of-Values Match in a Query

• A **list-of-values match** selects records whose value for the designated field matches one of two or more simple condition values
  – Could include several Or conditions in the design grid, but there is an easier and clearer way to do this
• The **In comparison operator** lets you define a condition with a list of two or more values for a field that select and include those record in the query results
Using the Not Logical Operator in a Query

- The **Not logical operator** negates a criterion or selects records for which the designated field does not match the criterion.
Using an AutoFilter to Filter Data Data

- You can use the AutoFilter feature to choose restrictions faster and with more flexibility than using the condition and Or logical operator.
Using an AutoFilter to Filter Data (Cont.)

Figure 5-7 Using an AutoFilter to filter records in the query recordset

- Toggles filter off and on
- Hartford and West Hartford patients selected
- Filtered indicator
Assigning a Conditional Value to a Calculated Field

• Null Fields are fields in a record does not contain any information at all

• A field in a record that contains any data at all—even a single space—is nonnull
  – To combine nonnull LastName and FirstName fields, you can use the expression LastName & “,” & FirstName

• The & *(ampersand)* operator is a concatenation operator that joins text expressions

• *Concatenation* refers to joining two or more text fields or characters encapsulated in quotes
Assigning a Conditional Value to a Calculated Field (Cont.)

- The IIf (Immediate If) function assigns one value to a calculated field or control if a condition is true, and a second value if the condition is false
  - The IIf function has three parts: a condition that is true or false, the result when the condition is true, and the result when the condition is false
  - Each part of the IIf function is separated by a comma
  - The IsNull function tests a field value or an expression for a null value; if the field value or expression is null, the result is true; otherwise, the result is false
Assigning a Conditional Value to a Calculated Field (Cont.)

Figure 5-8  If function inserted for the calculated field

- Calculated field name
- Inserted If function
- If function
Assigning a Conditional Value to a Calculated Field (Cont.)

Figure 5-9: After entering the condition for the calculated field's IIf function

- Condition for the IIf function
- ScreenTip for the IsNull function
- IsNull function
Assigning a Conditional Value to a Calculated Field  (Cont.)

Figure 5-10  Completed calculated field
Assigning a Conditional Value to a Calculated Field (Cont.)

Figure 5-11 Property sheet for the Patient calculated field

- Properties for the calculated field
- Calculated field is the current field
Assigning a Conditional Value to a Calculated Field (Cont.)

Figure 5-12  Completed query displaying the Patient calculated field

<table>
<thead>
<tr>
<th>Patient</th>
<th>Patient ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Parent</th>
<th>Date of Birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aguilar, Lilian</td>
<td>22504</td>
<td>Aguilar</td>
<td>Lilian</td>
<td></td>
<td>8/16/1938</td>
</tr>
<tr>
<td>Belanger, Malcolm</td>
<td>22519</td>
<td>Belanger</td>
<td>Malcolm</td>
<td></td>
<td>10/17/1950</td>
</tr>
<tr>
<td>Billings, Claire</td>
<td>22541</td>
<td>Billings</td>
<td>Claire</td>
<td></td>
<td>11/16/1990</td>
</tr>
<tr>
<td>Booker, Thomas</td>
<td>22510</td>
<td>Booker</td>
<td>Thomas</td>
<td></td>
<td>8/25/1966</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boucher, Sam</td>
<td>22543</td>
<td>Boucher</td>
<td>Sam</td>
<td></td>
<td>3/11/1975</td>
</tr>
<tr>
<td>Brown, Olivia</td>
<td>22530</td>
<td>Brown</td>
<td>Olivia</td>
<td></td>
<td>11/24/1943</td>
</tr>
<tr>
<td>Caputo, Michael</td>
<td>22536</td>
<td>Caputo</td>
<td>Michael</td>
<td></td>
<td>10/19/1998</td>
</tr>
<tr>
<td>Castro, Daniel</td>
<td>22511</td>
<td>Castro</td>
<td>Daniel</td>
<td></td>
<td>9/23/1933</td>
</tr>
<tr>
<td>Chang, Lisa</td>
<td>22512</td>
<td>Chang</td>
<td>Lisa</td>
<td></td>
<td>10/5/1955</td>
</tr>
<tr>
<td>Cruz, Magdalena</td>
<td>22550</td>
<td>Cruz</td>
<td>Magdalena</td>
<td></td>
<td>7/24/1984</td>
</tr>
<tr>
<td>Darcy, Edward</td>
<td>22501</td>
<td>Darcy</td>
<td>Edward</td>
<td></td>
<td>7/15/1986</td>
</tr>
<tr>
<td>Delgado, Alex</td>
<td>22535</td>
<td>Delgado</td>
<td>Alex</td>
<td></td>
<td>7/16/1980</td>
</tr>
<tr>
<td>Diaz, Anna</td>
<td>22542</td>
<td>Diaz</td>
<td>Anna</td>
<td></td>
<td>9/25/1987</td>
</tr>
<tr>
<td>Engber, Jim (Parent)</td>
<td>22521</td>
<td>Engber</td>
<td>Cathy</td>
<td>Engber, Jim</td>
<td>4/7/2006</td>
</tr>
<tr>
<td>Fielder, Pam</td>
<td>22549</td>
<td>Fielder</td>
<td>Pam</td>
<td></td>
<td>12/6/1978</td>
</tr>
<tr>
<td>Finnerty, Amber</td>
<td>22505</td>
<td>Finnerty</td>
<td>Amber</td>
<td></td>
<td>5/7/1946</td>
</tr>
<tr>
<td>Franklin, Chaney</td>
<td>22551</td>
<td>Franklin</td>
<td>Chaney</td>
<td></td>
<td>1/18/1954</td>
</tr>
<tr>
<td>Fraser, Nancy</td>
<td>22523</td>
<td>Fraser</td>
<td>Nancy</td>
<td></td>
<td>11/8/1977</td>
</tr>
<tr>
<td>Garrett, Ashley</td>
<td>22552</td>
<td>Garrett</td>
<td>Ashley</td>
<td></td>
<td>3/24/1989</td>
</tr>
</tbody>
</table>

- Patient names are the concatenation of LastName, FirstName for null Parent values.
- Patient names are the same as nonnull Parent values with the additional "(Parent)" text.
Creating a Parameter Query

- A **parameter query** displays a dialog box that prompts the user to enter one or more criteria values when the query is run.
  - The value entered into the prompt causes the query to select only those records with field value from the table.

![Figure 5-13: Specifying the prompt for the parameter query](image)
Creating a Parameter Query (Cont.)

**Figure 5-14** Enter Parameter Value dialog box

**Figure 5-15** Results of the parameter query
Creating a More Flexible Parameter Query

- Most users want a parameter query to display the records that match the parameter value the user enters or to display all records when the user doesn’t enter a parameter value
  - To provide this functionality, you can change the value in the Criteria box in the design grid for the specified column
  - Prefix the Like operator to the original criterion and concatenate the criterion to a wildcard character

![Modified City Criteria value in the Zoom dialog box](image)
Advanced Query Wizards

A cross-tab query uses aggregate functions such as Sum and Count to perform arithmetic operations on selected records.

A simple query selects records from one or more tables that satisfy criteria.

A find duplicates query is a select query that finds duplicate records in a table or query.

A find unmatched query is a select query that finds all records in a table or query that have no related records in a second table or query.

Find Unmatched Query Wizard

This find unmatched query will find all records that do not have matching records in both the tblVisit and tblBilling tables.

The tblVisit and tblBilling tables are joined on the VisitID field.

Find Duplicates Query Wizard

This find duplicates query will find records that have the same VisitDate field value.

The cross-tab query will display one column for the paid invoices and a second column for the unpaid invoices.

The selected field (InvoiceAmt) is used in the calculations for each column and row intersection.

This option determines whether to display an overall totals column in the cross-tab query.

The cross-tab query will display one row for each unique City field value.

Each column and row intersection will display the sum of the InvoiceAmt values.

The tblVisit and tblBilling tables are joined on the VisitID field.

New Perspectives on Microsoft Access 2013

25
Creating a Crosstab Query

Table: Aggregate functions used in crosstab queries

<table>
<thead>
<tr>
<th>Aggregate Function</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>Average of the field values</td>
</tr>
<tr>
<td>Count</td>
<td>Number of the nonnull field values</td>
</tr>
<tr>
<td>First</td>
<td>First field value</td>
</tr>
<tr>
<td>Last</td>
<td>Last field value</td>
</tr>
<tr>
<td>Max</td>
<td>Highest field value</td>
</tr>
<tr>
<td>Min</td>
<td>Lowest field value</td>
</tr>
<tr>
<td>StDev</td>
<td>Standard deviation of the field values</td>
</tr>
<tr>
<td>Sum</td>
<td>Total of the field values</td>
</tr>
<tr>
<td>Var</td>
<td>Variance of the field values</td>
</tr>
</tbody>
</table>

©2014 Cengage Learning
Creating a Crosstab Query (Cont.)

Figure 5-18

Comparing a select query to a crosstab query

- Results of a select query
- Individual West Hartford records
- Results of a crosstab query
- One row for West Hartford invoice amounts
- West Hartford record with unpaid invoice
- West Hartford records with paid invoices
- Paid invoices
- Unpaid invoices
Creating a Crosstab Query (Cont.)

The quickest way to create a crosstab query is to use the Crosstab Query Wizard.
Creating a Crosstab Query (Cont.)

**Figure 5-21**  
Crosstab query recordset

- unpaid invoices by city
- paid invoices by city

**Figure 5-22**  
Crosstab query in the design grid

- produces row headings
- produces column headings
- produces total values
- produces grand total column heading and values
Creating a Crosstab Query (Cont.)

Figure 5-23  IIf function for the crosstab query column headings

If (InvoicePaid, 'Paid', 'Unpaid')
Creating a Find Duplicates Query

• A find duplicates query is a select query that finds duplicate records in a table or query
  – You can create this type of query using the Find Duplicates Query Wizard

• A find duplicates query searches for duplicate values based on the fields you select when answering the Wizard’s questions
Creating a Find Unmatched Query

- A find unmatched query is a select query that finds all records in a table or query that have no related records in a second table or query
  - Use the **Find Unmatched Query Wizard** to create this type of query

![Diagram of Find Unmatched Query Wizard]

**Figure 5-25** Selecting the common field

- matching field in the tblVisit table
- matching field in the qryPatientsByNamedQuery
- click to confirm after selecting matching fields
- matching fields will appear here
Creating a Find Unmatched Query  (Cont.)

Figure 5-26  Query recordset displaying four patients without visits
Creating a Top Values Query

- Users might want to limit the number to a more manageable size by displaying, for example, just the first 10 records
  - The **Top Values property** for a query lets you limit the number of records in the query results
Creating a Top Values Query (Cont.)

Figure 5-28  Top values query recordset
Lookup Fields and Input Masks

The tblInvoiceItem query supplies the field values for the lookup field in the tblBilling table. A lookup field lets the user select a value from a list of possible values to enter data into the field.

The tblPatient table contains the field that displays values with an input mask. An input mask is a predefined format that is used to enter and display data in a field.

The Phone field uses an input mask to format displayed field values.

You can create an input mask for any field with the Short Text or Number data type.

The tblBilling table contains the lookup field.

The invoicetemID and invoicetem fields from the tblInvoiceItem table are used to look up invoicetemID values in the tblBilling table.

The Phone field uses an input mask to format displayed field values.

You can create an input mask for any field with the Short Text or Number data type.

The 9 character in an input mask indicates a digit or space in the field value whose entry is optional.

The \ indicates that the character that follows is a literal display character.

The character after the :: indicates what character to display as the user is entering data. In this case the \ will be displayed.

The 0 character in an input mask indicates that only a digit can be entered and the entry is mandatory.

Values in the lookup field appear in alphabetical order, sorted by Invoice Item.

Only the invoicetemID values are stored in the invoicetemID field in the tblBilling table even though the user also sees the invoicetem values in the datasheet.
Creating a Lookup Field

- Data entry is easier if users do not need to remember the correct values for fields
- A lookup field lets the user select a value from a list of possible values
- Use a **Lookup Wizard** field in Access to create a lookup field in a table
Creating a Lookup Field (Cont.)

Figure 5-30  Selecting the lookup fields

Figure 5-31  Adjusting the width of the lookup column
Creating a Lookup Field  (Cont.)

Figure 5-32  List of Invoiceltem and InvoiceltemID field values

<table>
<thead>
<tr>
<th>Invoice Num</th>
<th>Visit ID</th>
<th>Invoice Date</th>
<th>Invoice Amt</th>
<th>Invoice Item</th>
<th>Invoice Paid</th>
<th>Click to Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>38539</td>
<td>1544</td>
<td>11/30/2015</td>
<td>$85.00</td>
<td>Pharmacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38540</td>
<td>1544</td>
<td>11/30/2015</td>
<td>$48.00</td>
<td>IM Injection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38544</td>
<td>1548</td>
<td>12/01/2015</td>
<td>$100.00</td>
<td>Office visit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38545</td>
<td>1548</td>
<td>12/01/2015</td>
<td>$85.00</td>
<td>Pharmacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38548</td>
<td>1550</td>
<td>12/02/2015</td>
<td>$100.00</td>
<td>Office visit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38550</td>
<td>1550</td>
<td>12/02/2015</td>
<td>$250.00</td>
<td>Radiograph</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38552</td>
<td>1552</td>
<td>12/04/2015</td>
<td>$100.00</td>
<td>Pharmacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38554</td>
<td>1552</td>
<td>12/04/2015</td>
<td>$100.00</td>
<td>Office visit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Navigation Pane

scrollable list of values for the lookup field
Using the Input Mask Wizard

• A literal display character is a special character that automatically appears in specific positions of a field value – like hyphens in a social security number
  – To include these characters, you need to create an input mask, which is a predefined format used to enter and display data in a field

• An easy way to create an input mask is to use the Input Mask Wizard which guides you in creating a predefined format for a field
  – You must be in Design view to use the Input Mask Wizard
Using the Input Mask Wizard (Cont.)

Figure 5-33   Input Mask Wizard dialog box

scrollable list of predefined input masks

sample values for the corresponding input masks

practice area
Using the Input Mask Wizard (Cont.)

![Image showing the Input Mask Wizard with a phone number input mask example. The image highlights the build button and the input mask for a phone number.]

**Figure 5-34** Phone number input mask created by the Input Mask Wizard
Using the Input Mask Wizard  
(Cont.)

<table>
<thead>
<tr>
<th>Input Mask Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Digit only must be entered. Entry is required.</td>
</tr>
<tr>
<td>9</td>
<td>Digit or space can be entered. Entry is optional.</td>
</tr>
<tr>
<td>#</td>
<td>Digit, space, or a plus or minus sign can be entered. Entry is optional.</td>
</tr>
<tr>
<td>L</td>
<td>Letter only must be entered. Entry is required.</td>
</tr>
<tr>
<td>?</td>
<td>Letter only can be entered. Entry is optional.</td>
</tr>
<tr>
<td>A</td>
<td>Letter or digit must be entered. Entry is required.</td>
</tr>
<tr>
<td>a</td>
<td>Letter or digit can be entered. Entry is optional.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Any character or a space must be entered. Entry is required.</td>
</tr>
<tr>
<td>C</td>
<td>Any character or a space can be entered. Entry is optional.</td>
</tr>
<tr>
<td>&gt;</td>
<td>All characters that follow are displayed in uppercase.</td>
</tr>
<tr>
<td>&lt;</td>
<td>All characters that follow are displayed in lowercase.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Enclosed characters treated as literal display characters.</td>
</tr>
<tr>
<td>\</td>
<td>Following character treated as a literal display character. This is the same as enclosing a single character in quotation marks.</td>
</tr>
<tr>
<td>!</td>
<td>Input mask is displayed from right to left, rather than the default of left to right. Characters typed into the mask always fill in from left to right.</td>
</tr>
<tr>
<td>;;</td>
<td>The character between the first and second semicolons determines whether to store the literal display characters in the database. If the value is 1 or if no value is provided, the literal display characters are not stored. If the value is 0, the literal display characters are stored. The character following the second semicolon is the placeholder character that appears in the displayed input mask.</td>
</tr>
</tbody>
</table>

© 2014 Cengage Learning
Using the Input Mask Wizard (Cont.)

Figure 5-36  Property Update Options button menu
Using the Input Mask Wizard (Cont.)

Figure 5-37  Update Properties dialog box

Figure 5-38  After changing the Phone field input mask
Identifying Object Dependencies

• An **object dependency** exists between two objects when a change to the properties of data in one object affects the properties of data in the other object
  – Dependencies between Access objects, such as tables, queries, and forms, can occur as relationships or using a query to obtain values from more than one table.
  – Any form or report that uses fields from a query is directly dependent on the query and is indirectly dependent on the tables that provide the data to the query

• The **Object Dependencies pane** displays a collapsible list of the dependencies among the objects in an Access database
Identifying Object Dependencies (Cont.)

Figure 5-39 After opening the Object Dependencies pane

- Object Dependencies box
- Drag this edge to the left
- Warning messages and help
Defining Field Validation Rules

- To prevent a user from entering an unacceptable value in a field, you can create a **field validation rule** that verifies a field value by comparing it to a constant or to a set of constants.

  - You create a field validation rule by setting the **Validation Rule** and the **Validation Text** field properties.
  - The **Validation Rule property** value specifies the valid values that users can enter in a field.
  - The **Validation Text property** value will be displayed in a dialog box if a user enters an invalid value.
Defining Data Validation Rules (Cont.)

Figure 5-40 Validation properties for the InvoiceAmt field

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvoiceNum</td>
<td>Short Text</td>
</tr>
<tr>
<td>VisitID</td>
<td>Short Text</td>
</tr>
<tr>
<td>InvoiceDate</td>
<td>Date/Time</td>
</tr>
<tr>
<td>InvoiceAmt</td>
<td>Currency</td>
</tr>
<tr>
<td>InvoiceItemID</td>
<td>Short Text</td>
</tr>
<tr>
<td>InvoicePaid</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Insurance</td>
<td>Currency</td>
</tr>
</tbody>
</table>

- Current field: InvoiceAmt
- Validation properties:
  - Validation Rule: > 10
  - Validation Text: Invoice amounts must be greater than 10
  - Required: No
  - Indexed: No
  - Text Align: General
• **Defining Table Validation Rules**
  
  – To make sure that the value a user enters is not larger than the maximum field value, you can create a **table validation rule**
  
  – Use the Validation Rule and Validation Text properties and set these properties for the table instead of for an individual field
  
  – Use a table validation rule because this validation involves multiple fields
  
  – A field validation rule is used when the validation involves a restriction for only the selected field, and does not depend on other fields
Defining Data Validation Rules (Cont.)

Figure 5-41 Setting table validation properties

- Drag the edge to the left to widen the Property Sheet
- Properties for the table
- Validation properties
Working with Long Text Fields

• Use a Long Text field to store long comments and explanations

• Short Text fields are limited to 255 characters, but Long Text fields can hold up to 65,535 characters
  – Short Text fields limit you to plain text with no special formatting
  – Long Text fields store plain text similar to Short Text fields or to store rich text, which you can selectively format with options such as bold, italic, and different fonts and colors
Working with Long Text Fields (Cont.)

**Figure 5-42** Freezing three datasheet columns

scroll bar moved to the right, hiding the Reason/Diagnosis column
Working with Long Text Fields (Cont.)

Figure 5-43  Viewing the properties for a Long Text field

- Current field, a Long Text field
- Rich Text property setting
Designating a Trusted Folder

• A database is a file, and files can contain malicious instructions that can damage other files on your computer or files on other computers on your network

• Unless you take special steps, Access treats every database as a potential threat to your computer

• One special step that you can take is to designate a folder as a **trusted folder** which is a folder on a drive or network that you designate as trusted and where you place databases you know are safe
Designating a Trusted Folder (Cont.)

Figure 5-44  Designating a trusted folder

Image description:
- The Trust Center window is open, showing the User Locations section.
- Trusted folders listed here.
- Options for managing trusted folders.

New Perspectives on Microsoft Access 2013