

FIT1013 Digital Futures: IT for Business

Week 11 : Database Queries

DR. YEN CHEUNG

CLAYTON, FACULTY OF INFORMATION TECHNOLOGY
MONASH UNIVERSITY

On completion of your study this week, you should aim to:

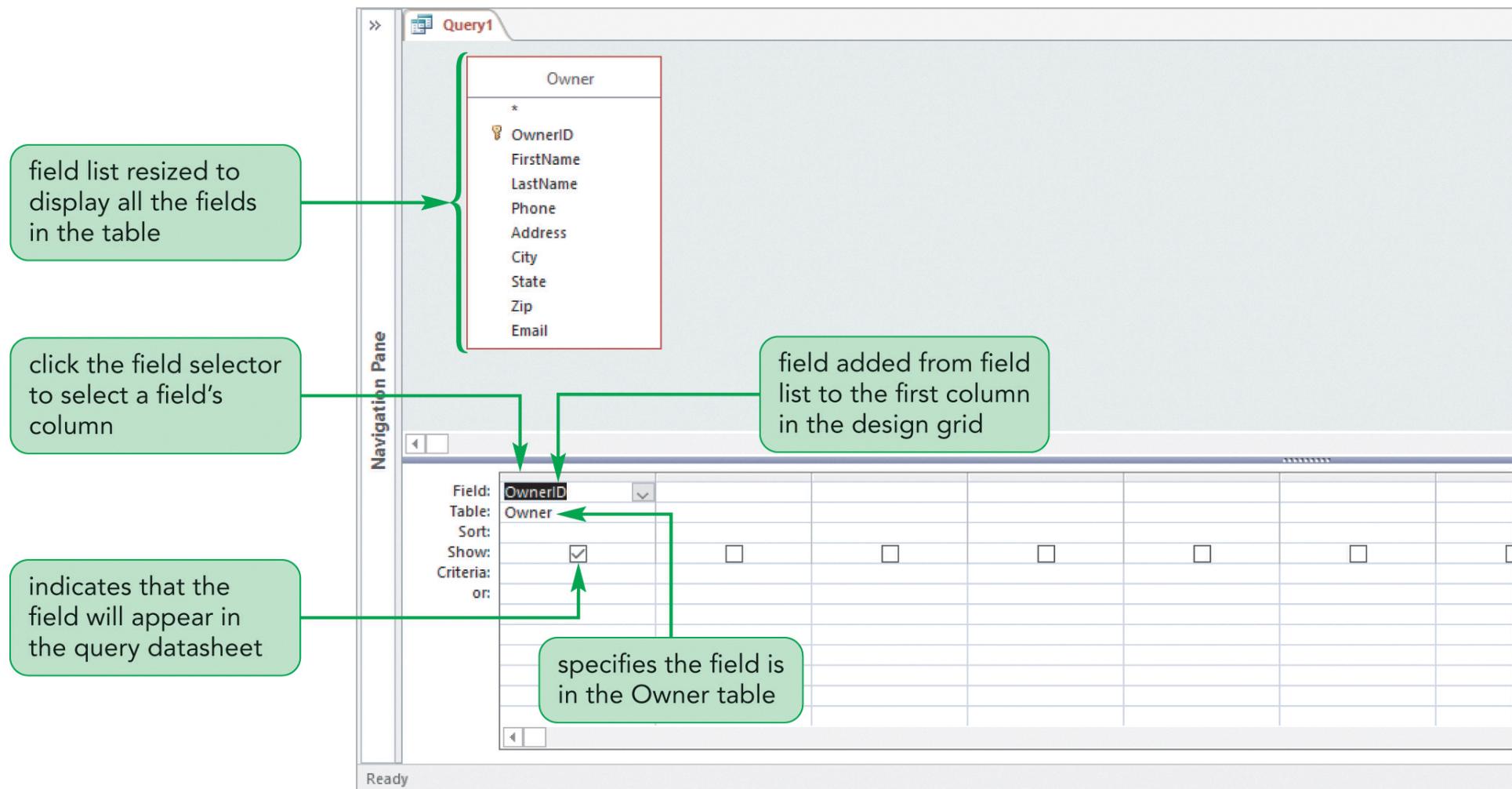
- Create a query based on multiple tables
- Use a comparison operator in a query to match a range of values
- Use the And and Or logical operators in queries
- Create and format a calculated field in a query
- Perform calculations in a query using aggregate functions and record group calculations
- Use the Like, In, Not, and & operators in queries
- Create a parameter query
- Use query wizards to create a crosstab query, a find duplicates query, and a find unmatched query
- Create a top values query



Introduction to Queries

- Access provides powerful query capabilities that allow you to do the following:
 - Display selected fields and records from a table
 - Sort records
 - Perform calculations
 - Generate data for forms, reports, and other queries
 - Update data in the tables in a database
 - Find and display data from two or more tables
- The answer to a select query is returned in the form of a datasheet
 - The result of a query is also referred to as a recordset because the query produces a set of records that answers your question

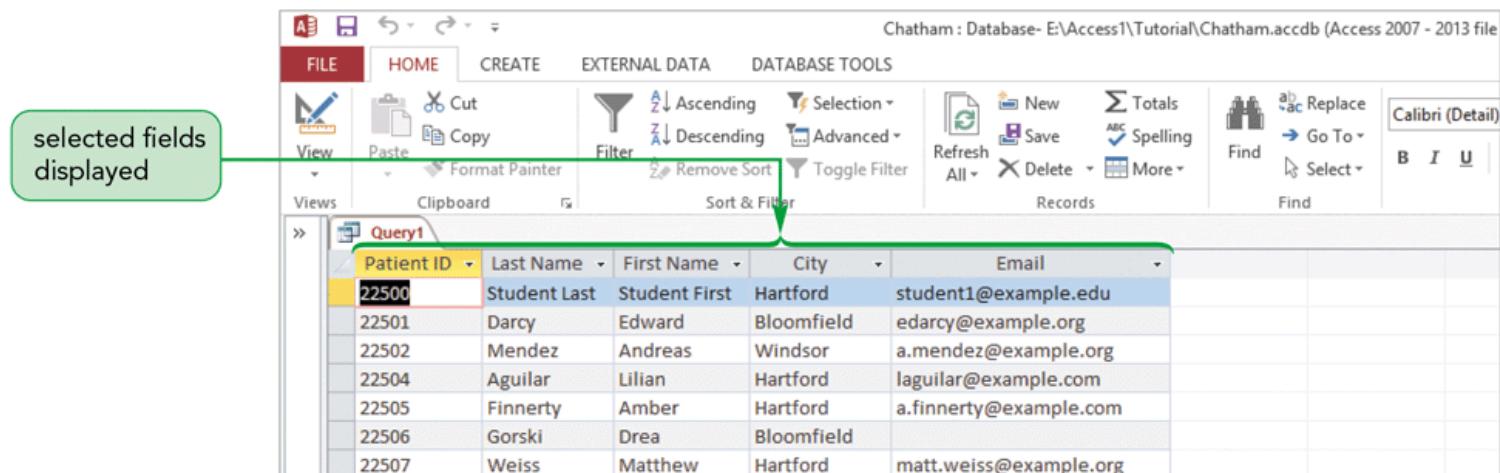
Creating and Running a Query



Creating and Running a Query (Cont.)

Figure 3-8

Datasheet displayed after running the query



A screenshot of the Microsoft Access application interface. The title bar reads "Chatham : Database- E:\Access1\Tutorial\Chatham.accdb (Access 2007 - 2013 file)". The ribbon tabs are FILE, HOME, CREATE, EXTERNAL DATA, and DATABASE TOOLS. The HOME tab is selected. The toolbar includes View, Cut, Copy, Paste, Filter, Sort & Filter, Refresh, New, Totals, Spelling, Find, and Replace. A green callout bubble points to the first row of the datasheet, which is highlighted in yellow. The first column header is "Patient ID". The data rows show various patient records with columns: Patient ID, Last Name, First Name, City, and Email.

Patient ID	Last Name	First Name	City	Email
22500	Student Last	Student First	Hartford	student1@example.edu
22501	Darcy	Edward	Bloomfield	edarcy@example.org
22502	Mendez	Andreas	Windsor	a.mendez@example.org
22504	Aguilar	Lilian	Hartford	laguilar@example.com
22505	Finnerty	Amber	Hartford	a.finnerty@example.com
22506	Gorski	Drea	Bloomfield	
22507	Weiss	Matthew	Hartford	matt.weiss@example.org

Creating a Multitable Query

- A multitable query is a query based on more than one table
- If you want to create a query that retrieves data from multiple tables, the tables must have a common field

Figure 3-9 Datasheet for query based on the Patient and Visit tables

City	First Name	Last Name	Date of Visit	Reason/Diagnosis
Hartford	Student First	Student Last	11/17/2015	Migraine
Bloomfield	Edward	Darcy	11/30/2015	Influenza
Windsor	Andreas	Mendez	3/30/2016	Annual wellness visit
Hartford	Lilian	Aguilar	11/18/2015	Annual wellness visit
Hartford	Amber	Finnerty	1/26/2016	Annual wellness visit
Bloomfield	Drea	Gorski	4/1/2016	Fifth Disease
Bloomfield	Drea	Gorski	4/8/2016	Fifth Disease follow-up
Hartford	Matthew	Weiss	11/9/2015	Diabetes mellitus Type 2 - initial diagnosis
Hartford	Matthew	Weiss	2/9/2016	Diabetes mellitus Type 2 - serum glucose check
Hartford	Matthew	Weiss	4/7/2016	Diabetes mellitus Type 2 - serum glucose check
Hartford	Steve	Kervin	4/4/2016	Tinea pedis
Hartford	Steve	Kervin	4/18/2016	Tinea pedis follow-up
Hartford	Thomas	Booker	11/10/2015	Seborrheic dermatitis
Hartford	Thomas	Booker	3/1/2016	Seborrheic dermatitis follow-up
West Hartford	Daniel	Castro	12/3/2015	Annual wellness visit
West Hartford	Daniel	Castro	1/13/2016	Cardiac monitoring
Hartford	Lisa	Chang	1/5/2016	Annual wellness visit
Bloomfield	Troy	Smith	1/13/2016	Broken leg
Bloomfield	Troy	Smith	2/24/2016	Follow-up - cast removal
Hartford	Ian	Parker	12/15/2015	Influenza
Hartford	Ian	Parker	1/14/2016	Hypertension monitoring
Hartford	Susan	King	12/22/2015	COPD management visit
West Hartford	Henry	O'Brien	2/1/2016	Annual wellness visit
West Hartford	Henry	O'Brien	4/11/2016	Idiopathic abdominal pain
Hartford	Sera	Torres	4/1/2016	Conjunctivitis

Selection Criteria in Queries

When creating queries in Design view, you can enter criteria so that Access will display only selected records in the query results.

Field:	PatientID	LastName	FirstName	BirthDate	City
Table:	Patient	Patient	Patient	Patient	Patient
Sort:					
Show:	<input checked="" type="checkbox"/>				
Criteria:					Bloomfield*
or:					

To define a condition for a field, you place the condition in the field's Criteria box in the design grid.

To tell Access which records you want to select, you must specify a condition as part of the query. A **condition** is a criterion, or rule, that determines which records are selected.

The results of a query containing selection criteria include only the records that meet the specified criteria.

Patient ID	Last Name	First Name	Date of Birth	City
22501	Darcy	Edward	7/15/1986	Bloomfield
22506	Gorski	Drea	2/19/2005	Bloomfield
22513	Srnith	Troy	1/31/1996	Bloomfield
22521	Engber	Cathy	4/7/2006	Bloomfield
22522	Li	Siyang	7/25/1986	Bloomfield
22549	Fielder	Fam	12/6/1978	Bloomfield
22557	Kirk	Isobel	11/18/1965	Bloomfield
*				

The results of this query show only patients from Bloomfield because the condition "Bloomfield" in the City field's Criteria box specifies that Access should select records only with City field values of Bloomfield. This type of condition is called an **exact match** because the value in the specified field must match the condition exactly in order for the record to be included in the query results.

A condition usually consists of an operator, often a comparison operator, and a value. A **comparison operator** asks Access to compare the value in a field to the condition value and to select all the records for which the condition is true.

Field:	InvoiceNum	InvoiceDate	InvoiceAmt	
Table:	Billing	Billing	Billing	
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			>250	
or:				

Invoice Num	Invoice Date	Invoice Amt
35815	11/16/2015	\$300.00
35900	01/20/2016	\$300.00
36002	03/15/2016	\$450.00
36074	04/12/2016	\$450.00
*		\$0.00

The results of this query show only those invoices with amounts greater than \$250 because the condition >250, which uses the greater than comparison operator, specifies that Access should select records only with InvoiceAmt field values over \$250.

Most comparison operators (such as Between...And...) ask Access to select records that match a range of values for the condition—in this case, all records with dates that fall within the range shown.

Field:	VisitID	PatientID	VisitDate	Reason
Table:	Visit	Visit	Visit	Visit
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Between #12/1/2015# And #12/31/2015#	
or:				

Visit ID	Patient ID	Date of Visit	Reason/Diagnosis
1550	22549	12/1/2015	Influenza
1552	22511	12/3/2015	Annual wellness visit
1555	22520	12/7/2015	Annual wellness visit
1557	22526	12/10/2015	Annual wellness visit
1560	22514	12/15/2015	Influenza
1562	22516	12/22/2015	COPD management visit
*			

The results of this query show only those patient visits that took place in December 2015 because the condition in the VisitDate's Criteria box specifies that Access should select records only with a visit date between 12/1/2015 and 12/31/2015.

Defining Record Selection Criteria for Queries

- To tell Access which records you want to select, you must specify a condition as part of the query
 - A condition usually includes one of the comparison operators

Figure 3-16

Access comparison operators

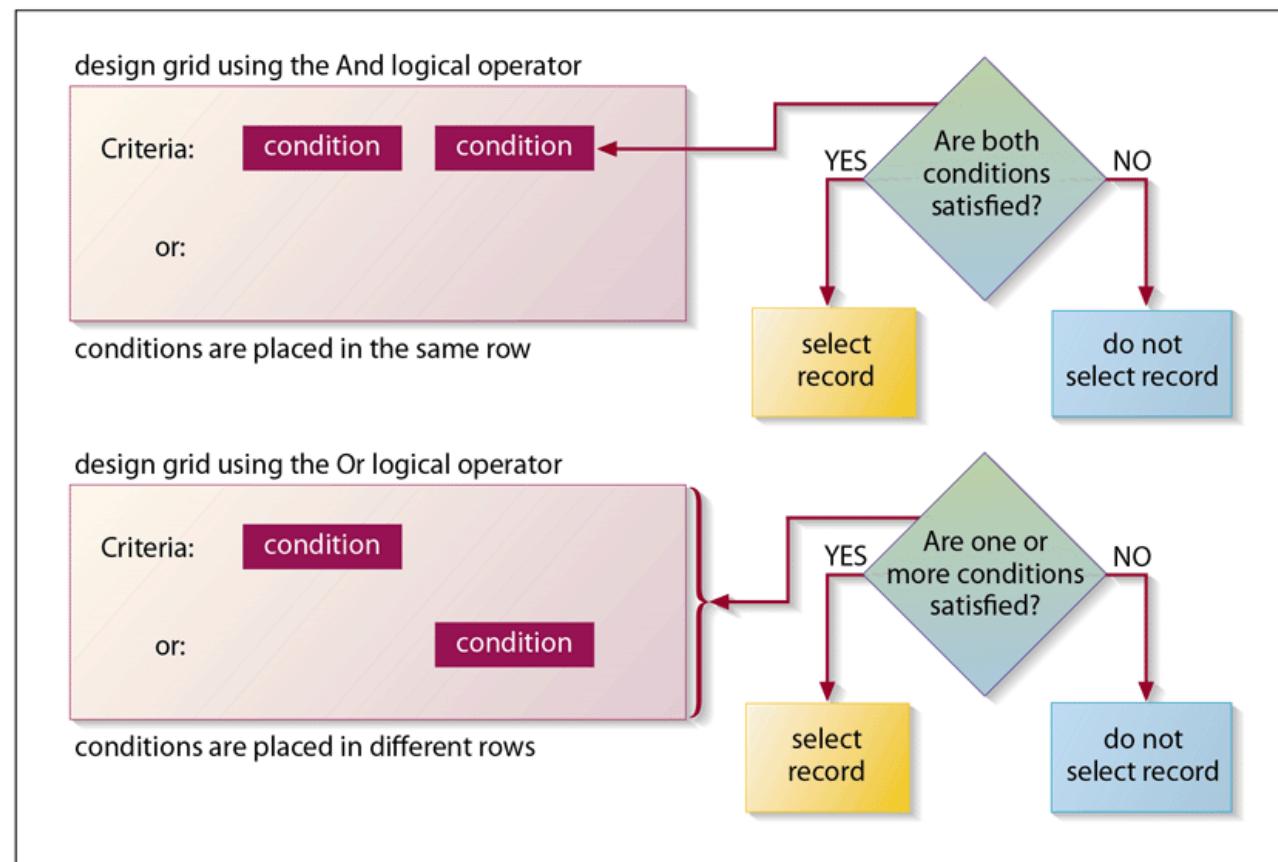
Operator	Meaning	Example
=	equal to (optional; default operator)	="Hall"
<>	not equal to	<>"Hall"
<	less than	<#1/1/99#
<=	less than or equal to	<=100
>	greater than	>"C400"
>=	greater than or equal to	>=18.75
Between ... And ...	between two values (inclusive)	Between 50 And 325
In ()	in a list of values	In ("Hall", "Seeger")
Like	matches a pattern that includes wildcards	Like "706**"

© 2014 Cengage Learning

Defining Multiple Selection Criteria for Queries (Cont.)

Figure 3-26

Logical operators And and Or for multiple selection criteria



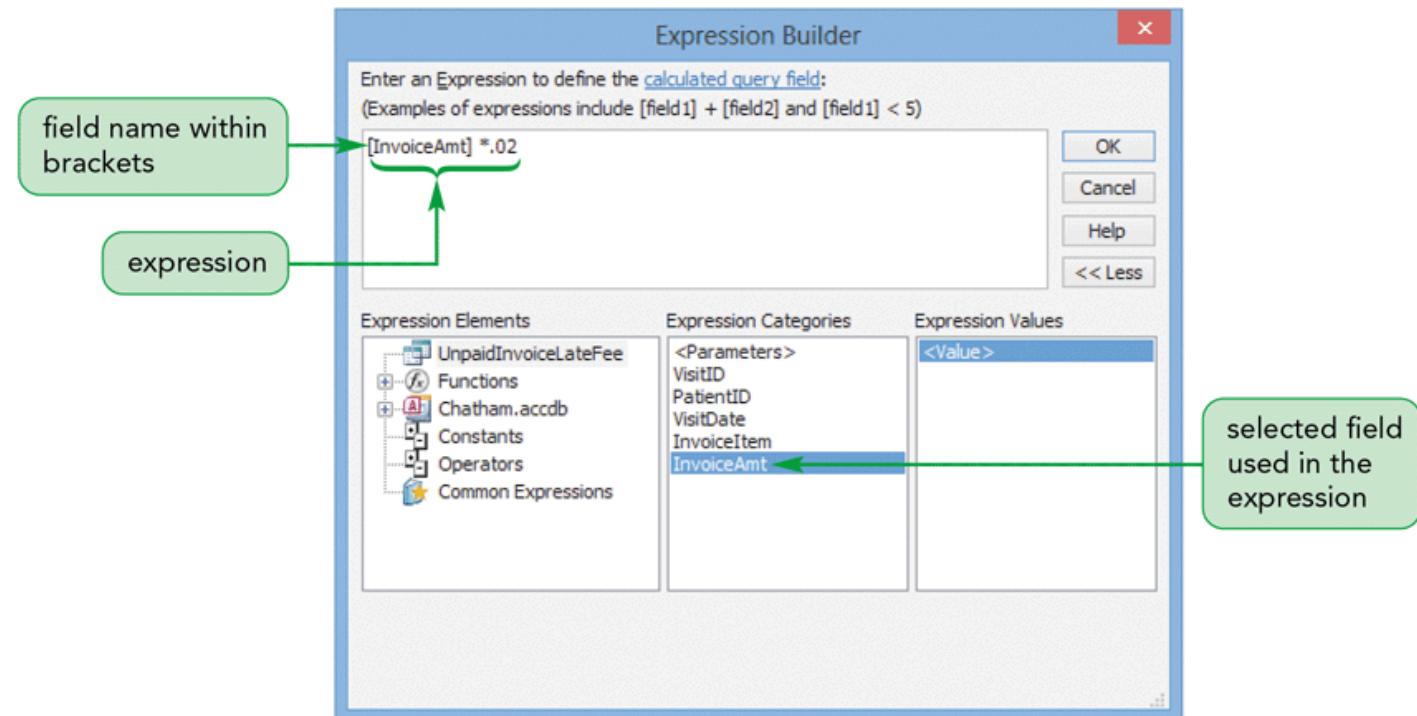
© 2014 Cengage Learning

Creating a Calculated Field

- Queries can perform calculations
 - Must define an **expression** containing a combination of database fields, constants, and operators
 - A **calculated field** is a field that displays the results of an expression but it does not exist in a database
 - The **Zoom box** is a dialog box that you can use to enter text, expressions, or other values
 - **Expression Builder** is an Access tool that makes it easy for you to create an expression
 - It contains a box for entering the expression, an option for displaying and choosing common operators, and one or more lists of expression elements, such as table and field names

Creating a Calculated Field (Cont.)

Figure 3-33 Completed expression for the calculated field



Using Aggregate Functions

- You can calculate statistical information, such as totals and averages, on the records displayed in a table datasheet or selected by a query
 - Use the Access **Aggregate functions** which perform arithmetic operations on selected records in a database

Figure 3-36 Frequently used aggregate functions

Aggregate Function	Determines	Data Types Supported
Average	Average of the field values for the selected records	AutoNumber, Currency, Date/Time, Number
Count	Number of records selected	AutoNumber, Currency, Date/Time, Long Text, Number, OLE Object, Short Text, Yes/No
Maximum	Highest field value for the selected records	AutoNumber, Currency, Date/Time, Number, Short Text
Minimum	Lowest field value for the selected records	AutoNumber, Currency, Date/Time, Number, Short Text
Sum	Total of the field values for the selected records	AutoNumber, Currency, Date/Time, Number

© 2014 Cengage Learning

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

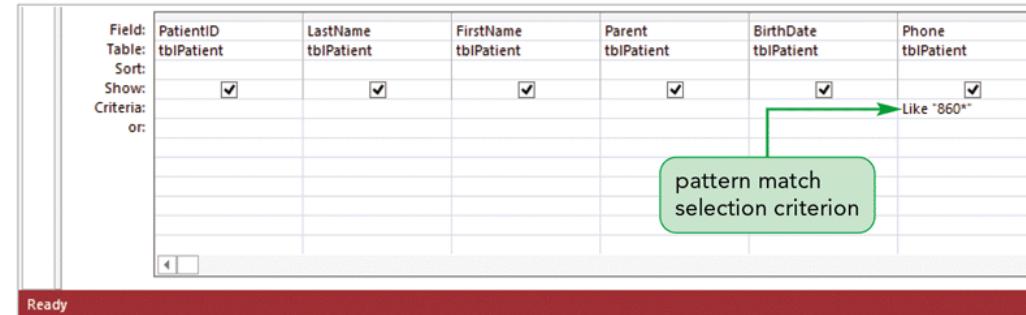


Figure 5-3 tblPatient table records for area code 860

The screenshot shows the Microsoft Access datasheet for the 'qry860AreaCode' query. The table has 13 columns: Patient ID, Last Name, First Name, Parent, Date of Birth, Phone, Address, City, State, Zip, and EmailAddress. The data consists of 46 rows, each representing a patient record. A green callout box in the bottom-left corner of the table area says '46 records total'. A green arrow points from this box to another green arrow on the right side of the table, which points to a green callout box containing the text 'scroll down to see more records that match the criteria'.

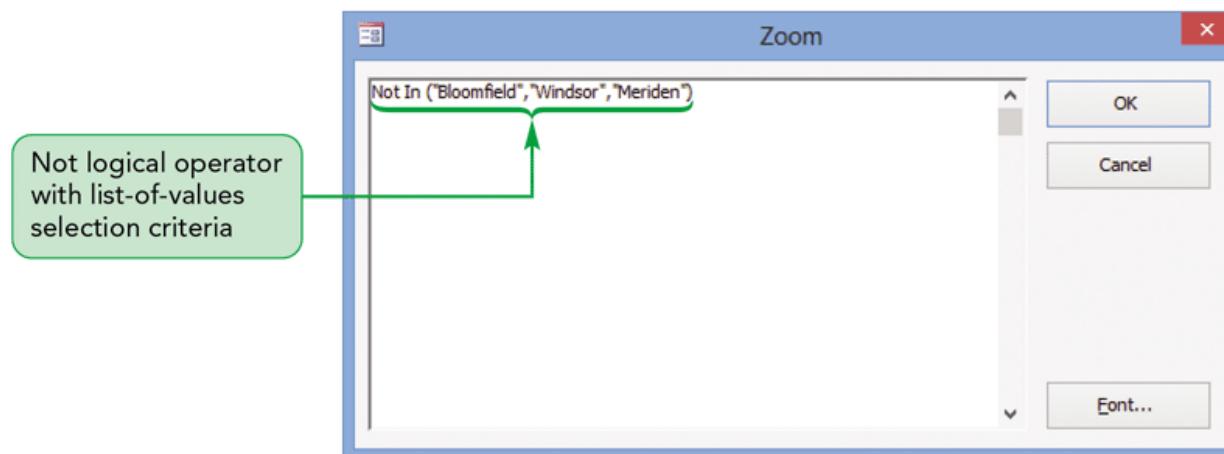
Patient ID	Last Name	First Name	Parent	Date of Birth	Phone	Address	City	State	Zip	EmailAddress
22500	Student Last	Student First		2/28/1994	8609382822	501 Perkins Dr	Hartford	CT	06120	student2@example.com
22501	Darcy	Edward		7/15/1986	8603053985	723 Oxford Ave	Bloomfield	CT	06002	edarcy@cengage.com
22502	Mendez	Andreas		3/1/1934	8605526893	57 Wallace Rd	Windsor	CT	06095	amendios@example.com
22504	Aguilar	Lillian		8/16/1938	8603745724	329 Fairchild Ave	Hartford	CT	06114	laguilar@cengage.com
22505	Finnerty	Amber		5/7/1946	8602264930	37 Noyes Ct	Hartford	CT	06112	amber glo@example.com
22506	Gorski	Drea	Samson, Ma	2/19/2005	8603058394	83 Everett Ln	Bloomfield	CT	06002	
22507	Weiss	Matthew	Weiss, Jordi	6/7/1997	8604267492	38 German St	Hartford	CT	06106	matt.weiss@cengage.com
22509	Kervin	Steve		4/6/1993	8609380025	49 Davenport St	Hartford	CT	06120	skervin@example.com
22511	Castro	Daniel		9/23/1933	8606370430	61 Osmond Way	West Hartford	CT	06117	d Castro@cengage.com
22512	Chang	Lisa		10/5/1955	8602266034	731 Macon Rd	Hartford	CT	06112	lchang14@example.com
22513	Smith	Troy		1/31/1996	8603050384	16 Ravine Rd	Bloomfield	CT	06002	troy.smith9@example.net
22514	Parker	Ian		6/3/1958	8609381873	12 Adelbert St	Hartford	CT	06120	ian_parker@example.com
22517	O'Brien	Henry		12/10/1940	8606379203	58 Redmond Dr	West Hartford	CT	06117	hobrien32@example.com
22518	Torres	Sera	Torres, Gina	4/9/2008	8609382098	27 Reno Dr	Hartford	CT	06120	serat483@cengage.com
22519	Belanger	Malcolm		10/17/1950	8606373927	723 Nicola Ave	West Hartford	CT	06117	
22520	Hallick	Davis		3/26/1944	8605524495	84 Churchill Pl	Windsor	CT	06095	dhallick@example.com
22521	Engber	Cathy	Engber, Jim	4/7/2006	8603053048	58 Deering Pl	Bloomfield	CT	06002	cengber2@cengage.com
22522	Li	Siyang		7/25/1986	8603056548	225 Krauss Rd	Bloomfield	CT	06002	lisiy3@example.com
22523	Fraser	Nancy		11/8/1977	8605527392	7 Quinn Dr	Windsor	CT	06095	
22526	Swenson	Lucia		5/1/1943	8602260293	83 Osage Ave	Hartford	CT	06112	
22527	Lee	Hwan		8/25/1987	8606613974	153 Agnes Ct	Hartford	CT	06105	hwan.lee@example.org
22529	Coldham	Robert		5/30/1961	8605522873	92 Gaston Ave	Windsor	CT	06095	
22534	Jessica			2/28/1980	8603742987	74 Brayton Dr	Hartford	CT	06114	jesswallner@example.net
22535	Christina			12/5/1941	8605525920	27 Tracey Ct	Windsor	CT	06095	crowe black@example.net
22536	Delgado	Alex		7/16/1960	8605521739	48 Warwick St	Hartford	CT	06105	susan.hawes4@example.com
	Caputo	Michael		10/19/1998	8603749347	96 Vega Dr	Hartford	CT	06114	

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

Figure 5-5

Record selection based on not matching a list of values



Assigning a Conditional Value to a Calculated Field (Cont.)

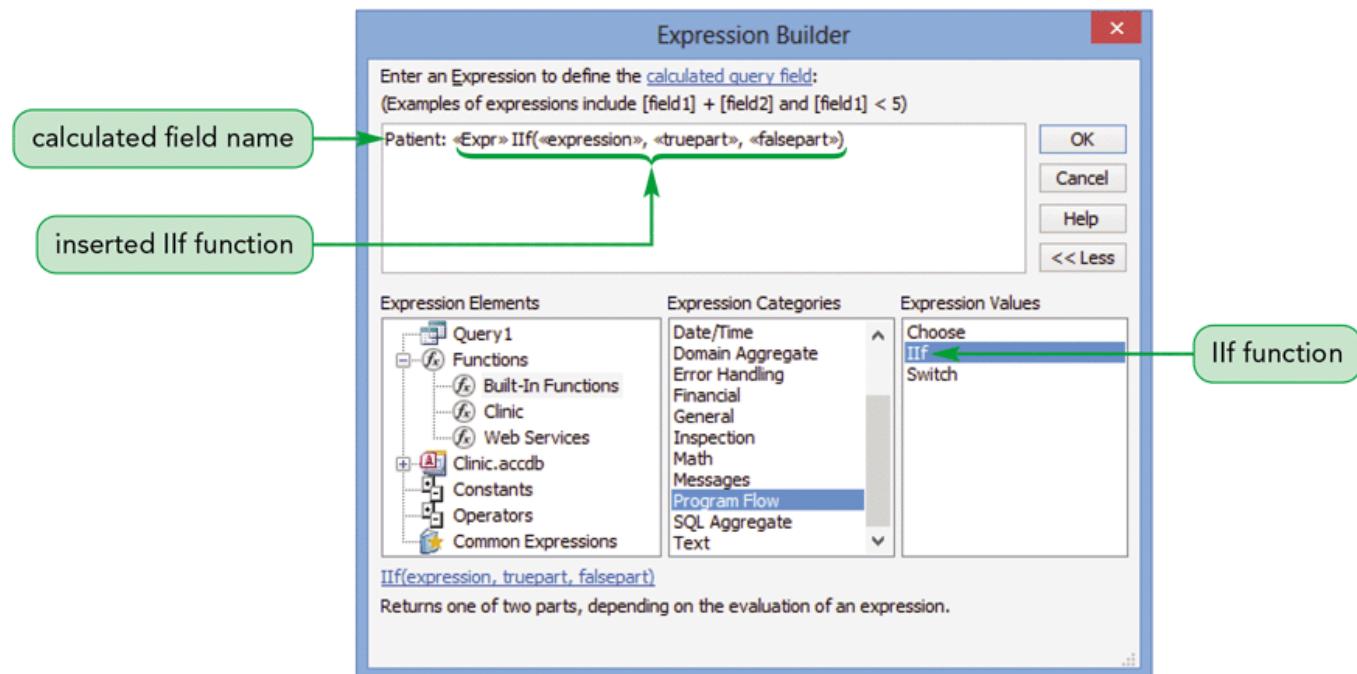
- The **IIf (Immediate If)** function assigns one value to a calculated field or control if a condition is true, and a second value if the condition is false
 - The **IIf function** has three parts: a condition that is true or false, the result when the condition is true, and the result when the condition 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

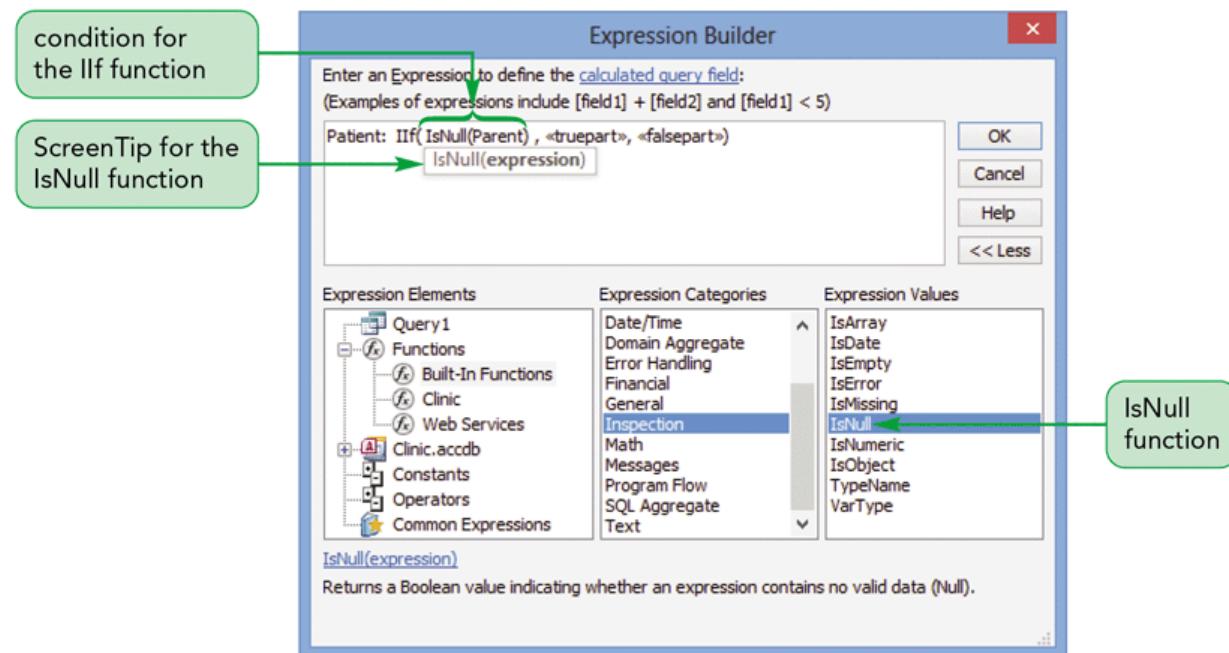
IIf function inserted for the calculated field



Assigning a Conditional Value to a Calculated Field

(Cont.)

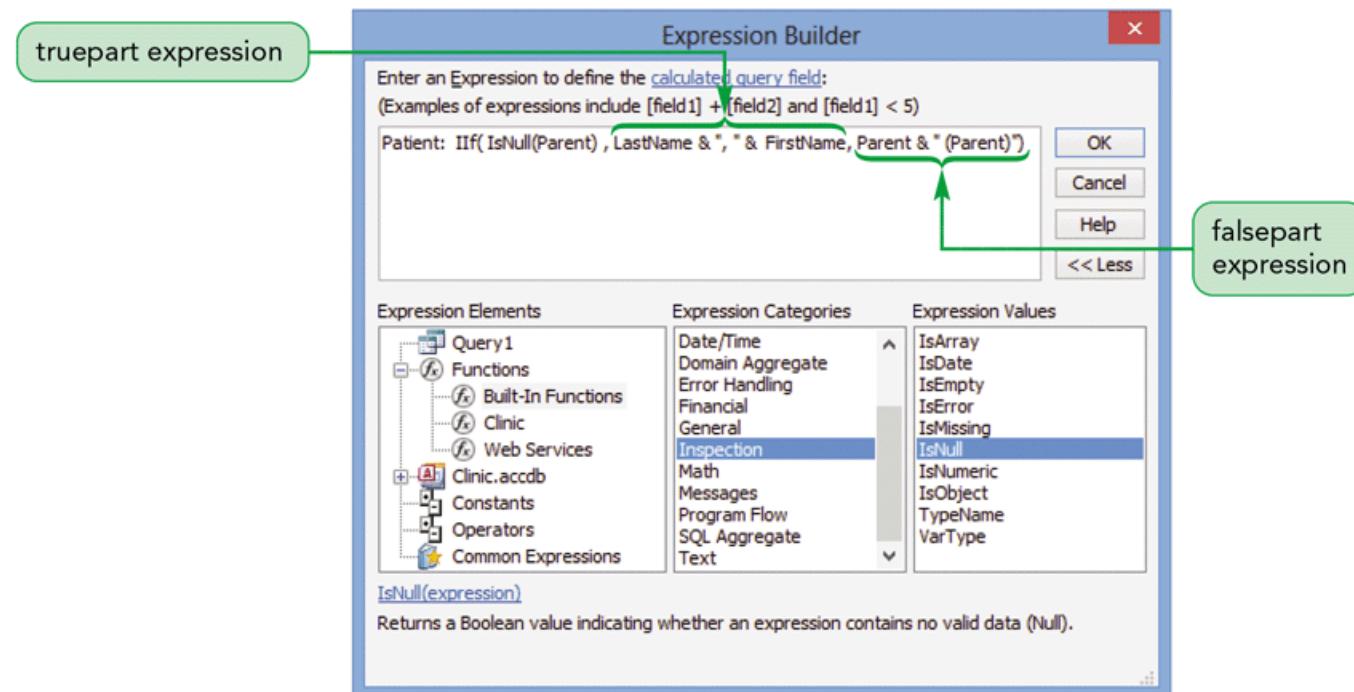
Figure 5-9 After entering the condition for the calculated field's IIf function



Assigning a Conditional Value to a Calculated Field

(Cont.)

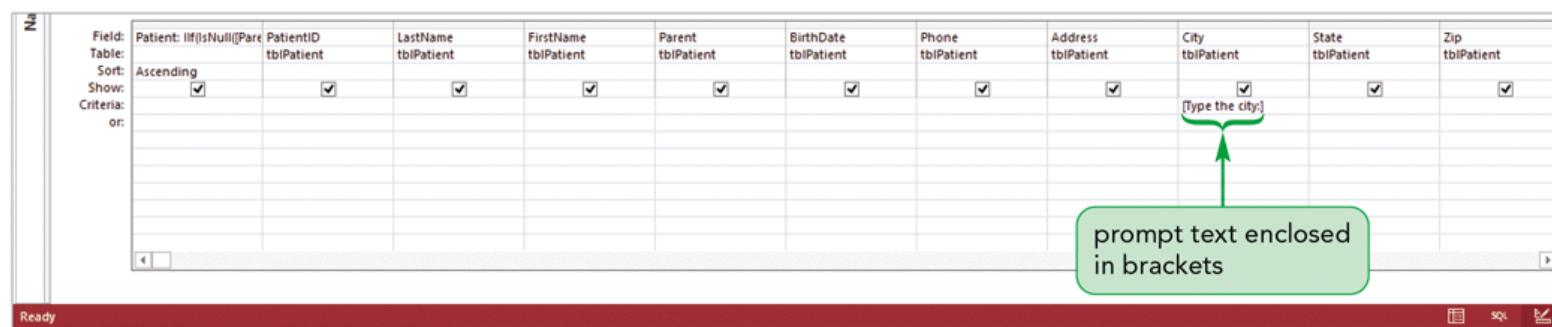
Figure 5-10 Completed calculated field



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



Creating a Parameter Query (Cont.)

Figure 5-14 Enter Parameter Value dialog box

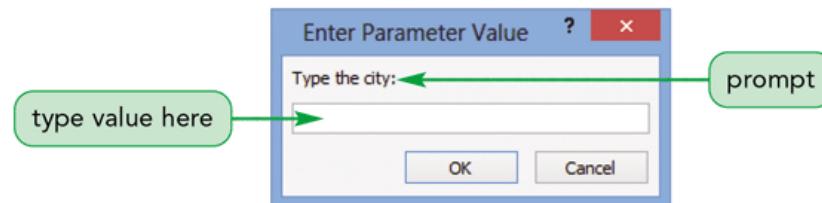


Figure 5-15 Results of the parameter query

A screenshot of Microsoft Access showing the results of a query named 'qryPatientsByCityParameter'. The table has columns: Patient, Patient ID, Last Name, First Name, Parent, Date of Birth, Phone, Address, City, State, Zip, and EmailAddress. Two rows are visible: one for 'Garrett, Ashley' and one for 'Rodriguez, Maria'. Both rows have 'Waterbury' listed under 'City' and 'CT' listed under 'State'. A green callout bubble labeled 'Waterbury patients selected' with an arrow points to the 'City' column for the first row.

Patient	Patient ID	Last Name	First Name	Parent	Date of Birth	Phone	Address	City	State	Zip	EmailAddress
Garrett, Ashley	22552	Garrett	Ashley		3/24/1989	4755528429	372 Higbee Ct	Waterbury	CT	06704	agarrett@example.org
Rodriguez, Maria	22525	Rodriguez	Maria		2/11/1936	4755529023	624 Noyes St	Waterbury	CT	06704	marodrig13@example.net

Advanced Query Wizards

The screenshot displays four Microsoft Access wizards:

- New Query:** Shows options for Simple Query Wizard, Crosstab Query Wizard, Find Duplicates Query Wizard, and Find Unmatched Query Wizard.
- Crosstab Query Wizard:** Used to calculate totals for multiple columns and rows. Fields: LastName, FirstName, InvoiceAmt. Functions: Avg, Count, First, Last, Max, Min, StDev, Sum, Var. Sample output shows a grid for City (City1, City2, City3, City4) with columns for InvoicePaid1, InvoicePaid2, and InvoicePaid3, each showing the sum of InvoiceAmt.
- Find Unmatched Query Wizard:** Used to find records in one table that don't have matches in another. Fields in 'tblVisit': VisitID, PatientID, VisitDate, Reason, WalkIn. Fields in 'tblBilling': InvoiceNum, VisitID, InvoiceDate, InvoiceAmt, InvoiceItemID, InvoicePaid, Insurance. Matching fields: VisitID <=> VisitID.
- Find Duplicates Query Wizard:** Used to find records with duplicate values in specified fields. Available fields: VisitID, PatientID, Reason, WalkIn. Duplicate-value fields: VisitDate.

Annotations provide additional context:

- New Query:**
 - A **crosstab 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.
- Crosstab Query Wizard:**
 - Each column and row intersection will display the sum of the InvoiceAmt values.
 - 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 crosstab query.
 - The crosstab query will display one column for the paid invoices and a second column for the unpaid invoices.
 - The crosstab query will display one row for each unique City field value.
- 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.



Creating a Crosstab Query

Figure 5-17 Aggregate functions used in crosstab queries

Aggregate Function	Definition
Avg	Average of the field values
Count	Number of the nonnull field values
First	First field value
Last	Last field value
Max	Highest field value
Min	Lowest field value
StDev	Standard deviation of the field values
Sum	Total of the field values
Var	Variance of the field values

©2014 Cengage Learning

Creating a Crosstab Query (Cont.)

Figure 5-18 Comparing a select query to a crosstab query

Last Name	First Name	City	Invoice Amt	Invoice Paid
Lewis	Patrice	Windsor	\$85.00	<input checked="" type="checkbox"/>
Lewis	Patrice	Windsor	\$48.00	<input checked="" type="checkbox"/>
Belanger	Malcolm	West Hartford	\$100.00	<input type="checkbox"/>
Darcy	Edward	Bloomfield	\$100.00	<input checked="" type="checkbox"/>
Darcy	Edward	Bloomfield	\$85.00	<input checked="" type="checkbox"/>
Fielder	Pam	Bloomfield	\$100.00	<input checked="" type="checkbox"/>
Fielder	Pam	Bloomfield	\$85.00	<input type="checkbox"/>
Fielder	Pam	Bloomfield	\$250.00	<input type="checkbox"/>
Castro	Daniel	West Hartford	\$100.00	<input checked="" type="checkbox"/>
Hallick	Davis	Windsor	\$100.00	<input type="checkbox"/>
Hallick	Davis	Windsor	\$72.00	<input type="checkbox"/>
Swenson	Lucia	Hartford	\$100.00	<input checked="" type="checkbox"/>
Swenson	Lucia	Hartford	\$85.00	<input checked="" type="checkbox"/>
Swenson	Lucia	Hartford	\$45.00	<input type="checkbox"/>
Swenson	Lucia	Hartford	\$32.00	<input type="checkbox"/>
Parker	Ian	Hartford	\$100.00	<input checked="" type="checkbox"/>
Parker	Ian	Hartford	\$85.00	<input checked="" type="checkbox"/>
King	Susan	Meriden	\$100.00	<input type="checkbox"/>
King	Susan	Meriden	\$150.00	<input type="checkbox"/>
Ingram	Julia	Hartford	\$100.00	<input checked="" type="checkbox"/>
Ingram	Julia	Hartford	\$75.00	<input type="checkbox"/>
Chang	Lisa	Hartford	\$100.00	<input checked="" type="checkbox"/>
Chang	Lisa	Hartford	\$85.00	<input checked="" type="checkbox"/>
Belanger	Malcolm	West Hartford	\$100.00	<input checked="" type="checkbox"/>
Ropiak	Jane	Hartford	\$100.00	<input checked="" type="checkbox"/>
Ropiak	Jane	Hartford	\$150.00	<input checked="" type="checkbox"/>
Shaw	Daniel	West Hartford	\$100.00	<input checked="" type="checkbox"/>

City	Total Of InvoiceAmt	Paid	Unpaid
Bloomfield	\$2,953.00	\$1,273.00	\$1,680.00
Hartford	\$9,040.00	\$4,681.00	\$4,359.00
Meriden	\$1,336.00	\$748.00	\$588.00
Waterbury	\$528.00	\$460.00	\$68.00
West Hartford	\$2,153.00	\$1,395.00	\$758.00
Windsor	\$4,693.00	\$3,591.00	\$1,102.00

Creating a Crosstab Query (Cont.)

- The quickest way to create a crosstab query is to use the **Crosstab Query Wizard**

Figure 5-19 Choosing the query for the crosstab query

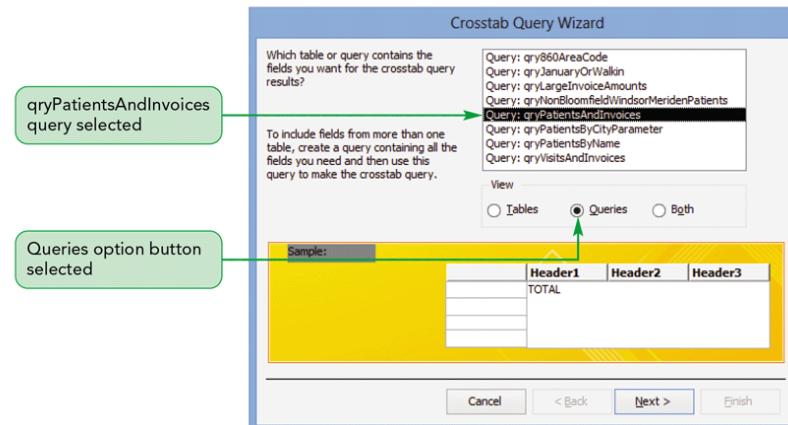
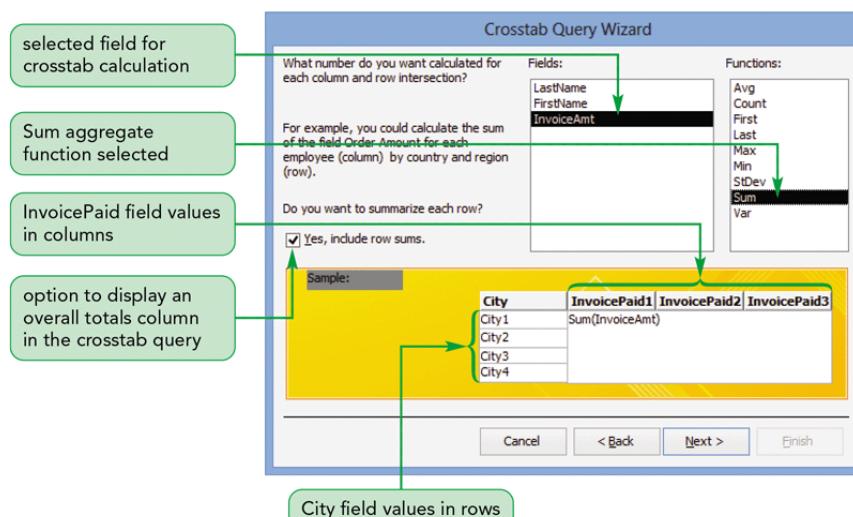


Figure 5-20 Completed crosstab query design



Creating a Crosstab Query (Cont.)

Figure 5-21 Crosstab query recordset

The screenshot shows the Microsoft Access interface with the 'CREATE' tab selected. The ribbon also includes 'HOME', 'EXTERNAL DATA', and 'DATABASE TOOLS'. The 'Queries' section of the ribbon is highlighted. A table titled 'qryPatientsAndInvoicesCrosstab' is displayed, showing data grouped by city. The columns represent 'City', 'Total Of InvoiceAmt', and three additional columns with values -\$1,273.00, \$1,680.00, and \$1,273.00 respectively. Two green callout boxes on the left side of the table point to the first two columns: 'unpaid invoices by city' points to the first column, and 'paid invoices by city' points to the second column.

City	Total Of InvoiceAmt	-1	0
Bloomfield	\$2,953.00	\$1,273.00	\$1,680.00
Hartford	\$9,040.00	\$4,681.00	\$4,359.00
Meriden	\$1,336.00	\$748.00	\$588.00
Waterbury	\$528.00	\$460.00	\$68.00
West Hartford	\$2,153.00	\$1,395.00	\$758.00
Windsor	\$4,693.00	\$3,591.00	\$1,102.00

Figure 5-22 Crosstab query in the design grid

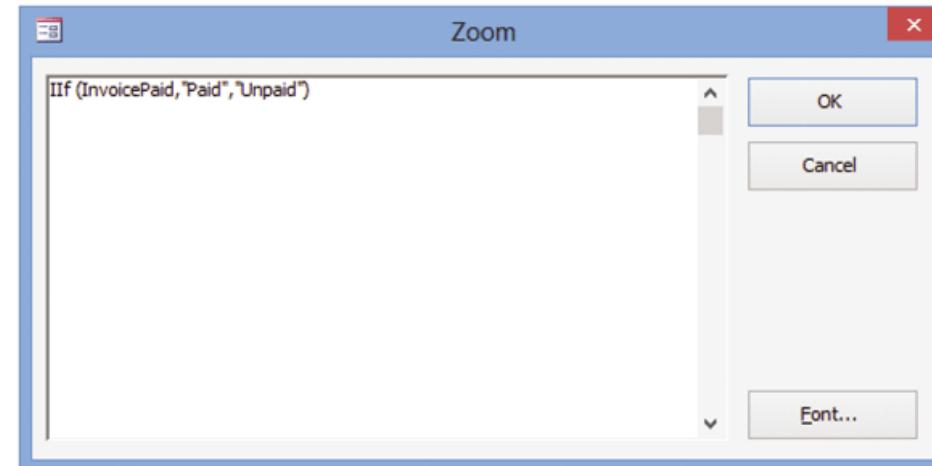
The screenshot shows the Microsoft Access 'Design View' for a Crosstab query. The 'Navigation' pane on the left lists fields: [City], [InvoicePaid], [InvoiceAmt], and 'Total Of InvoiceAmt: [InvoiceAmt]'. The main grid shows the query structure with four columns. Arrows from callout boxes on the left indicate the purpose of each column: the first column '[City]' is labeled 'produces row headings'; the second column '[InvoicePaid]' is labeled 'produces column headings'; the third column '[InvoiceAmt]' is labeled 'produces total values'; and the fourth column 'Total Of InvoiceAmt: [InvoiceAmt]' is labeled 'produces grand total column heading and values'.

Field:	[City]	Table:	qryPatientsAndInvoices	Total:	[InvoicePaid]	Crosstab:	[InvoiceAmt]	Group By	qryPatientsAndInvoices	Group By	[InvoiceAmt]	Sum	Value	Total Of InvoiceAmt: [InvoiceAmt]	qryPatientsAndInvoices
Criteria:	or:	Crossstab:	Sort:	Row Heading	Column Heading	Sort:	Row Heading	Criteria:	or:	Criteria:	or:	Criteria:	or:	Criteria:	or:

Creating a Crosstab Query (Cont.)

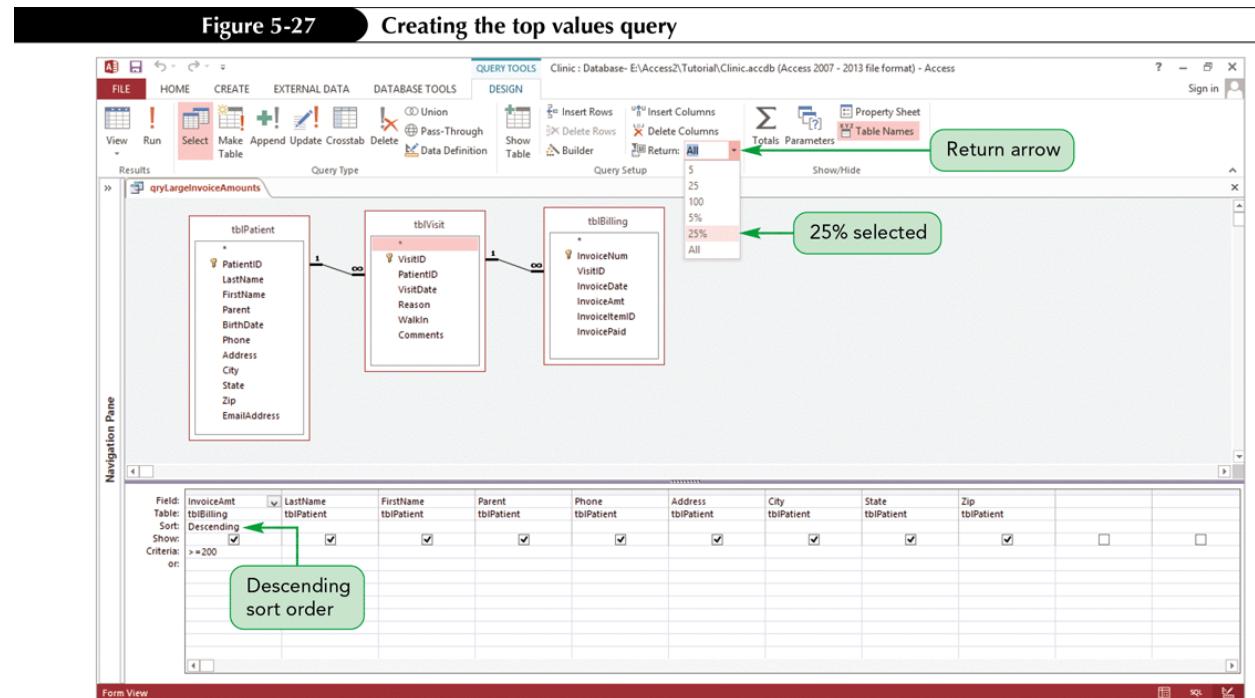
Figure 5-23

If function for the crosstab query column headings



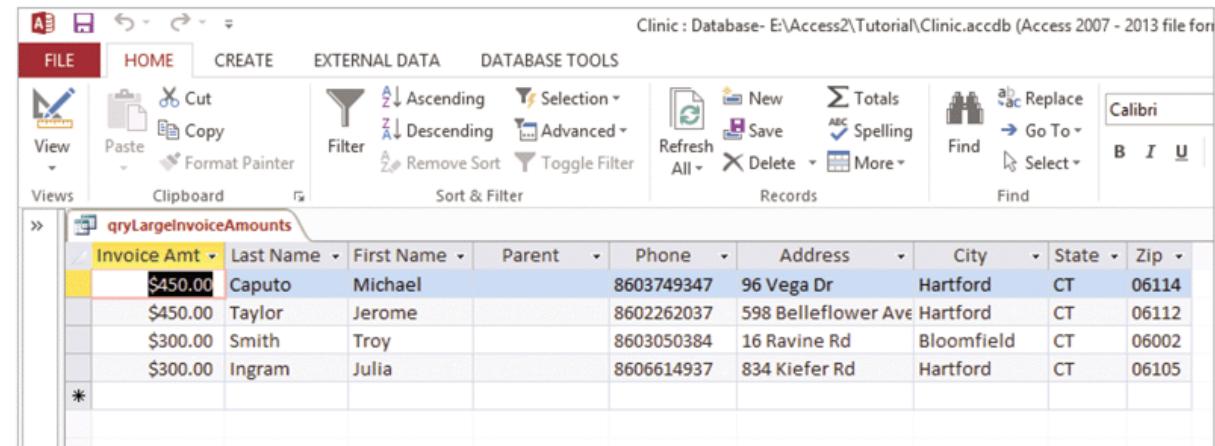
Creating a Top Values Query

- 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



The screenshot shows a Microsoft Access window displaying a query named "qryLargeInvoiceAmounts". The query results are shown in a table with the following data:

Invoice Amt	Last Name	First Name	Parent	Phone	Address	City	State	Zip
\$450.00	Caputo	Michael		8603749347	96 Vega Dr	Hartford	CT	06114
\$450.00	Taylor	Jerome		8602262037	598 Belleflower Ave	Hartford	CT	06112
\$300.00	Smith	Troy		8603050384	16 Ravine Rd	Bloomfield	CT	06002
\$300.00	Ingram	Julia		8606614937	834 Kiefer Rd	Hartford	CT	06105
*								

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.

Invoice Item ID	Invoice Item
DG111	Lab work
DG115	Lab work - culture
DG118	Lab work - glycated hemoglobin (A1C)
DG119	Lab work - urine glucose
DG225	Lab - culture
DG287	Lab - serum glucose
DG424	EKG with interpretation
DG532	Radiograph
OST145	Bone setting and cast
OST150	Cast of fracture

The tblBilling table contains the lookup field.

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

Invoice Num	Visit ID	Invoice Date	Invoice Amt	Invoice Item	Invoice Paid	Insurance
35801	1527	11/10/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	\$50.00
35802	1528	11/10/2015	\$100.00	Lab - culture	<input type="checkbox"/>	\$0.00
35803	1528	11/10/2015	\$45.00	Lab - serum glucose	<input type="checkbox"/>	\$0.00
35804	1528	11/13/2015	\$238.00	EKG with interpretation	<input type="checkbox"/>	\$0.00
35805	1528	11/13/2015	\$48.00	Radiograph	<input type="checkbox"/>	\$0.00
35808	1530	11/12/2015	\$100.00	Bone setting and cast	<input type="checkbox"/>	\$0.00
35809	1530	11/12/2015	\$85.00	Cast of fracture	<input type="checkbox"/>	\$0.00
35810	1530	11/12/2015	\$65.00	Cast removal	<input type="checkbox"/>	\$0.00
35811	1530	11/13/2015	\$48.00	Pharmacy	<input type="checkbox"/>	\$0.00
35813	1535	11/13/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	\$0.00
35814	1535	11/13/2015	\$45.00	IM injection	<input type="checkbox"/>	\$0.00
35815	1535	11/16/2015	\$300.00	Physical therapy	<input type="checkbox"/>	\$0.00
35816	1535	11/16/2015	\$250.00	Phlebotomy	<input type="checkbox"/>	\$0.00
35818	1536	11/18/2015	\$100.00	Influenza vaccine	<input type="checkbox"/>	\$0.00
35819	1536	11/18/2015	\$65.00	Respiratory therapy	<input type="checkbox"/>	\$100.00
35821	1538	11/18/2015	\$100.00	Surgery	<input type="checkbox"/>	\$0.00
35822	1538	11/18/2015	\$125.00	Suture removal	<input type="checkbox"/>	\$0.00
35825	1539	11/19/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	\$0.00

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

Only the InvoiceItemID values are stored in the InvoiceItemID field in the tblBilling table even though the user also sees the InvoiceItem values in the datasheet.

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.

Field Name	Data Type
PatientID	Short Text
LastName	Short Text
FirstName	Short Text
Parent	Short Text
BirthDate	Date/Time
Phone	Short Text
Address	Short Text
City	Short Text
State	Short Text
Zip	Short Text
EmailAddress	Short Text

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.

Field Properties

General	Lookup
Field Size	14
Format	999\,-00\,-0000;;_
Input Mask	999\,-00\,-0000;;_
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

Creating a Lookup Field (Cont.)

Figure 5-32

List of InvoiceItem and InvoiceItemID field values

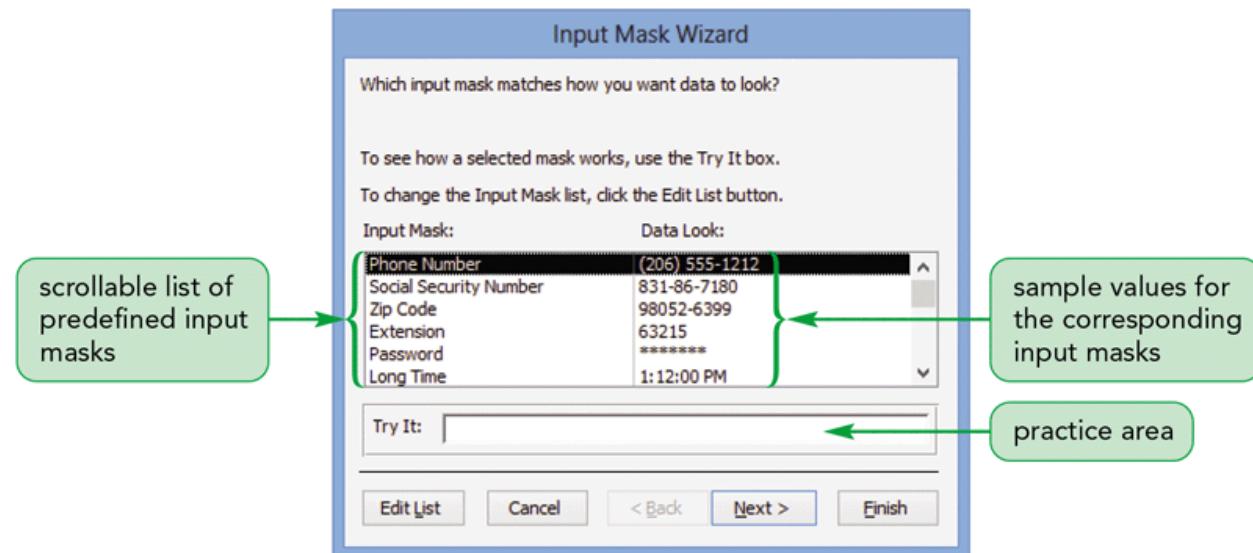
A screenshot of a Microsoft Access database window titled "tblBilling". The main grid shows columns for Invoice Num, Visit ID, Invoice Date, Invoice Amt, Invoice Item, Invoice Paid, and a "Click to Add" button. The "Invoice Item" column for row 35854 contains the value "Office visit". A dropdown menu is open over this cell, listing various medical services with their corresponding codes. A green callout bubble points to the top of the dropdown menu with the text "scrollable list of values for the lookup field".

Invoice Num	Visit ID	Invoice Date	Invoice Amt	Invoice Item	Invoice Paid	Click to Add
35839	1544	11/30/2015	\$85.00	Pharmacy	<input checked="" type="checkbox"/>	
35840	1544	11/30/2015	\$48.00	IM injection	<input checked="" type="checkbox"/>	
35844	1548	12/01/2015	\$100.00	Office visit	<input type="checkbox"/>	
35847	1549	12/01/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	
35848	1549	12/01/2015	\$85.00	Pharmacy	<input checked="" type="checkbox"/>	
35850	1550	12/02/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	
35851	1550	12/02/2015	\$85.00	Pharmacy	<input type="checkbox"/>	
35852	1550	12/04/2015	\$250.00	Radiograph	<input type="checkbox"/>	
35854	1552	12/04/2015	\$100.00	Office visit	<input checked="" type="checkbox"/>	
35857	1555	12/08/2015	\$100.00	Lab - culture	DG225	^
35858	1555	12/08/2015	\$72.00	Lab - serum glucose	DG287	
35860	1557	12/11/2015	\$100.00	EKG with interpretation	DG424	
35861	1557	12/11/2015	\$85.00	Radiograph	DG532	
35862	1557	12/11/2015	\$45.00	Bone setting and cast	OST145	
35863	1557	12/14/2015	\$32.00	Cast of fracture	OST150	
35865	1560	12/16/2015	\$100.00	Cast removal	OST158	
35866	1560	12/16/2015	\$85.00	Pharmacy	PRM712	
35868	1562	12/23/2015	\$100.00	Office visit	REP001	
35869	1562	12/28/2015	\$150.00	IM injection	REP139	
35872	1563	01/05/2016	\$100.00	Physical therapy	REP187	
35873	1563	01/05/2016	\$75.00	Phlebotomy	REP298	
35875	1564	01/06/2016	\$100.00	Influenza vaccine	REP725	
35876	1564	01/06/2016	\$85.00	Respiratory therapy	REP752	
35879	1567	01/11/2016	\$100.00	Surgery	SUR001	
35880	1569	01/12/2016	\$100.00	Suture removal	SUR145	▼
35881	1569	01/15/2016	\$150.00	Office visit	<input checked="" type="checkbox"/>	
35884	1570	01/12/2016	\$100.00	Respiratory therapy	<input checked="" type="checkbox"/>	
				Office visit	<input checked="" type="checkbox"/>	

Using the Input Mask Wizard (Cont.)

Figure 5-33

Input Mask Wizard dialog box



Using the Input Mask Wizard (Cont.)

Figure 5-34 Phone number input mask created by the Input Mask Wizard

The screenshot shows the Microsoft Access 'Design' view for a table named 'tblPatient'. The 'Phone' field is selected, highlighted with a red border. A callout bubble labeled 'input mask for a phone number' points to the 'Input Mask' field in the 'Field Properties' pane, which contains the value '(999) 0000-0000'. Another callout bubble labeled 'build button' points to the ellipsis (...) button in the same pane. The 'Field Properties' pane also lists other field properties like 'Field Size' (14), 'Format' (General), and 'Required' (No). The 'General' tab is selected in the properties pane.

Field Name	Data Type	Description (Optional)
PatientID	Short Text	Primary key
LastName	Short Text	
FirstName	Short Text	
Parent	Short Text	Parent or Guardian
BirthDate	Date/Time	
Phone	Short Text	
Address	Short Text	
City	Short Text	
State	Short Text	
Zip	Short Text	
EmailAddress	Short Text	

Navigation Pane

Field Properties

General Lookup

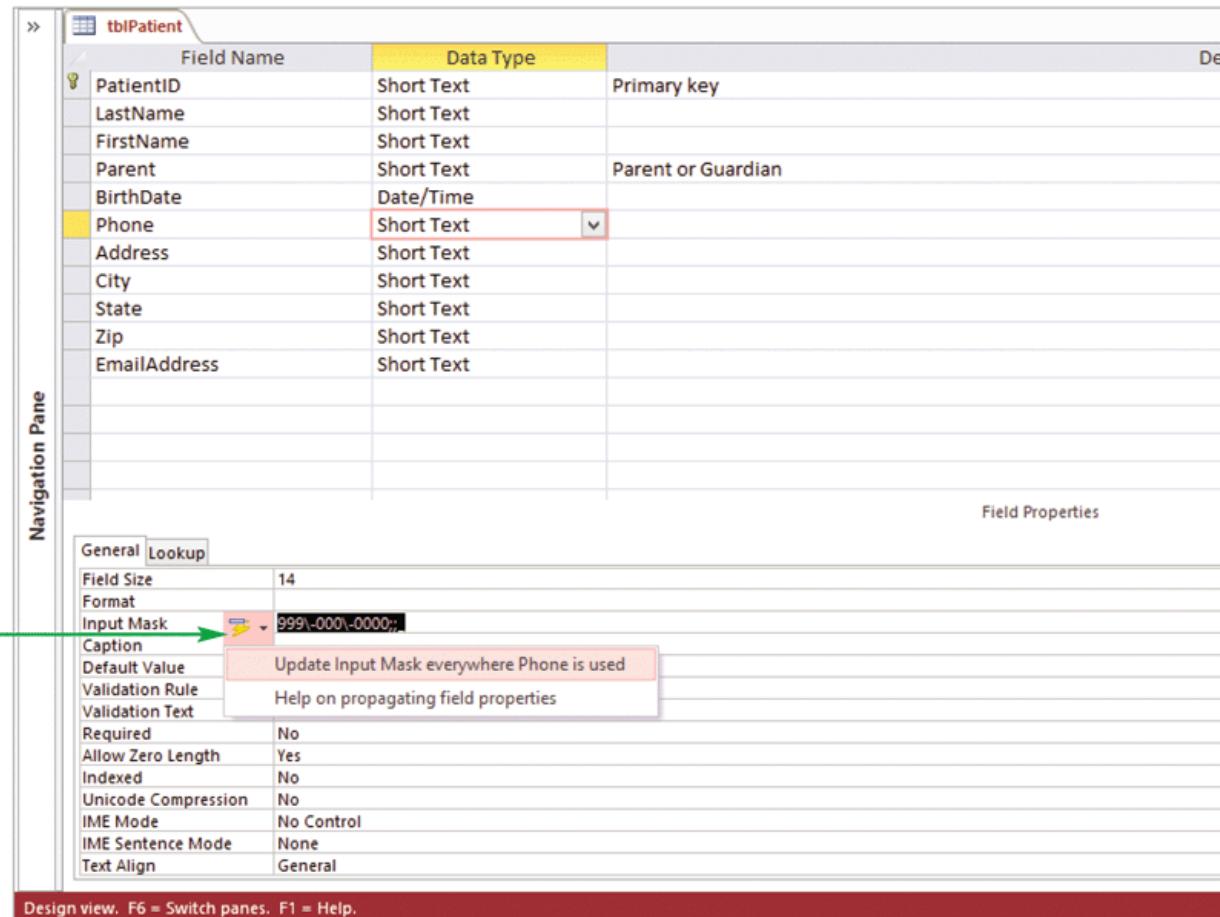
Field Size: 14
Format: General
Input Mask: (999) 0000-0000
Caption:
Default Value:
Validation Rule:
Validation Text:
Required: No
Allow Zero Length: Yes
Indexed: No
Unicode Compression: No
IME Mode: No Control
IME Sentence Mode: None
Text Align: General

A pattern for all data to be entered in this field

Design view. F6 = Switch panes. F1 = Help.

Using the Input Mask Wizard (Cont.)

Figure 5-36 Property Update Options button menu



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

The screenshot shows the Microsoft Access ribbon with the 'DATABASE TOOLS' tab selected. A table named 'tblPatient' is open in the foreground, displaying patient data. An 'Object Dependencies' pane is open on the right side of the window. The pane title is 'Object Dependencies' and it shows a tree view of dependencies for the 'tblPatient' table. A green callout bubble points to the edge of the pane with the text 'drag this edge to the left'. Another green callout bubble points to the bottom of the pane with the text 'warning messages and help'.

drag this edge to the left

Object Dependencies

Table: tblPatient

Objects that depend on me

Objects that I depend on

Tables

- tblVisit

Queries

- qry860AreaCode
- qryJanuaryOrWalkin
- qryLargeInvoiceAmounts
- qryNonBloomfieldWindsorMeridenPatients
- qryPatientsAndInvoices
- qryPatientsByCityParameter
- qryPatientsByName

Forms

- frmPatient

Reports

- None

Ignored Objects

- Unsupported Objects

Query: qryDuplicateVisitDates

Help

WARNING: Some objects were ignored

Things that cause dependencies

Primary key

Patient ID	Last Name	First Name	Parent	Date of Birth	Phone	Address	City	State	Zip	student
22500	Student Last	Student First		2/28/1994	860-938-2822	501 Perkins Dr	Hartford	CT	06120	student
22501	Darcy	Edward		7/15/1986	860-305-3985	723 Oxford Ave	Bloomfield	CT	06002	edard
22502	Mendez	Andreas		3/1/1934	860-552-6893	57 Wallace Rd	Windsor	CT	06095	amer
22504	Aguilar	Lillian		8/16/1938	860-374-5724	329 Fairchild Ave	Hartford	CT	06114	laguil
22505	Finnerty	Amber		5/7/1946	860-226-4930	37 Noyes Ct	Hartford	CT	06112	ambe
22506	Gorski	Drea	Samson, Mary	2/19/2005	860-305-8394	83 Everett Ln	Bloomfield	CT	06002	drea
22507	Weiss	Matthew	Weiss, Jordan	6/7/1997	860-426-7492	38 German St	Hartford	CT	06106	matt
22509	Kervin	Steve		4/6/1993	860-938-0025	49 Davenport St	Hartford	CT	06120	skerv
22510	Booker	Thomas		8/25/1966	203-661-2539	59 Nicola Ave	Meriden	CT	06450	
22511	Castro	Daniel		9/23/1933	860-637-0430	61 Osmond Way	West Hartford	CT	06117	d_cas
22512	Chang	Lisa		10/5/1955	860-226-6034	724 Main St	Hartford	CT	06112	lchan
22513	Smith	Troy		1/31/1996	860-305-0384	734 Main St	Hartford	CT	06112	troy
22514	Parker	Ian		6/3/1958	860-938-1873	74 Main St	Hartford	CT	06105	ian_p
22516	King	Susan		12/7/1963	203-661-9347	48 Fenton St	Meriden	CT	06450	sueki
22517	O'Brien	Henry		12/10/1940	860-637-9203	58 Redmond Dr	West Hartford	CT	06117	hobri
22518	Torres	Sera	Torres, Gina	4/9/2008	860-938-2098	27 Reno Dr	Hartford	CT	06120	serat
22519	Belanger	Malcolm		10/17/1950	860-637-3927	723 Nicola Ave	West Hartford	CT	06117	
22520	Hallick	Davis		3/26/1944	860-552-4495	84 Churchill Pl	Windsor	CT	06095	dhall
22521	Engber	Cathy	Engber, Jim	4/7/2006	860-305-3048	58 Deering Pl	Bloomfield	CT	06002	cengl
22522	Li	Siyang		7/25/1986	860-305-6548	225 Krauss Rd	Bloomfield	CT	06002	lisiy3
22523	Fraser	Nancy		11/8/1977	860-552-7392	7 Quinn Dr	Windsor	CT	06095	
22525	Rodriguez	Maria		2/11/1936	475-552-9023	624 Noyes St	Waterbury	CT	06704	maro
22526	Swenson	Lucia		5/1/1943	860-226-0293	83 Osage Ave	Hartford	CT	06112	
22527	Lee	Hwan		8/25/1987	860-661-3974	153 Agnes Ct	Hartford	CT	06105	hwar
22529	Goldberg	Robert		5/30/1961	860-552-2873	92 Gaston Ave	Hartford	CT	06112	
22530	Brown	Olivia		11/24/1943	203-938-7482	649 Scovill Dr	Hartford	CT	06112	
22531	Wallner	Jessica		2/28/1980	860-374-2987	74 Brayton Dr	Hartford	CT	06105	jesswi

Defining Data Validation Rules (Cont.)

▪ 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

Figure 5-40 Validation properties for the InvoiceAmt field

The screenshot shows the Microsoft Access Field Properties dialog for the 'InvoiceAmt' field in the 'tblBilling' table. The 'General' tab is selected, showing validation properties. A green callout labeled 'current field' points to the 'InvoiceAmt' row in the table above. Another green callout labeled 'validation properties' points to the 'Validation Rule' and 'Validation Text' fields in the 'General' tab.

Field Name	Data Type	
InvoiceNum	Short Text	Primary key
VisitID	Short Text	Foreign key
InvoiceDate	Date/Time	
InvoiceAmt	Currency	
InvoiceItemID	Short Text	
InvoicePaid	Yes/No	
Insurance	Currency	

General Tab Properties (for InvoiceAmt):

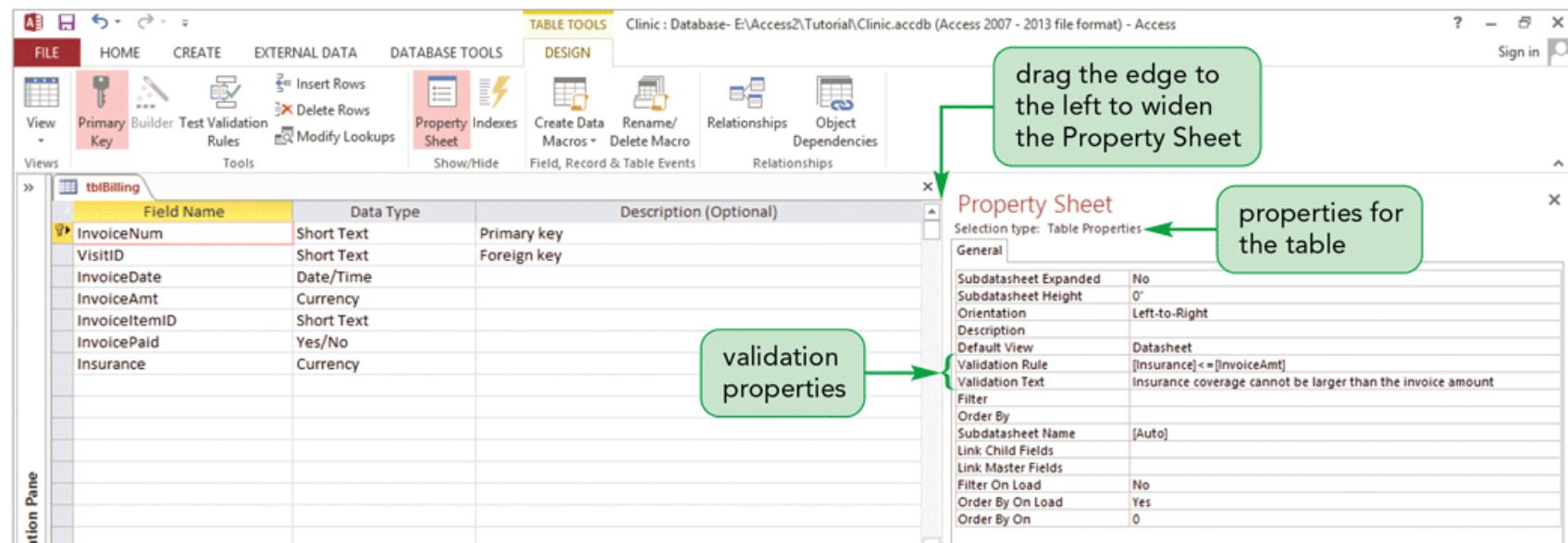
Format	Currency
Decimal Places	2
Input Mask	
Caption	Invoice Amt
Default Value	
Validation Rule	>10
Validation Text	Invoice amounts must be greater than 10
Required	No
Indexed	No
Text Align	General

Design view. F6 = Switch panes. F1 = Help.

Defining Data Validation Rules (Cont.)

Figure 5-41

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

The screenshot shows the Microsoft Access 'Design view' for a table named 'tblVisit'. The 'Field Name' column lists fields: VisitID, PatientID, VisitDate, Reason, Walkin, and Comments. The 'Data Type' column shows VisitID, PatientID, VisitDate, and Reason as 'Short Text', Walkin as 'Yes/No', and Comments as 'Long Text'. The 'Description (Optional)' column notes that 'Comments' contains 'Clinician observations'. A green callout box labeled 'current field, a Long Text field' points to the 'Comments' row. The 'Field Properties' pane is open, showing the 'General' tab selected. Under 'Text Format', the value 'Rich Text' is highlighted with a green arrow and a callout box labeled 'Rich Text property setting'. A note in the pane explains: 'Choose Rich text to store text as HTML and allow rich formatting. Choose Plain text to store only text. Avoid using Rich text if data might be used in Microsoft Access 2003 or earlier.' The status bar at the bottom left indicates 'Design view. F6 = Switch panes. F1 = Help.'

Field Name	Data Type	Description (Optional)
VisitID	Short Text	Primary key
PatientID	Short Text	Foreign key
VisitDate	Date/Time	Date of Visit
Reason	Short Text	
Walkin	Yes/No	
Comments	Long Text	Clinician observations

Navigation Pane

Field Properties

General Lookup

Format
Caption
Default Value
Validation Rule
Validation Text
Required
Allow Zero Length
Indexed
Unicode Compression
IME Mode
IME Sentence Mode
Text Format
Text Align
Append Only

No
Yes
No
Yes
No Control
None
Rich Text
Plain Text
Rich Text

Choose Rich text to store text as HTML and allow rich formatting. Choose Plain text to store only text. Avoid using Rich text if data might be used in Microsoft Access 2003 or earlier.

Design view. F6 = Switch panes. F1 = Help.

Summary

- Create different types of queries based on multiple tables
- Use operators in queries
- Create and format a calculated field in a query
- Perform calculations in a query

Homework

- Go through Access Tutorials 3 and 5
- Read Resources in Moodle

