MONASH University

FIT1013 Digital Futures: IT for Business
Week 11 : Database Queries

CLAYTON, FACULTY OF INFORMATION TECHNOLOGY

MONASH UNIVERSITY

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

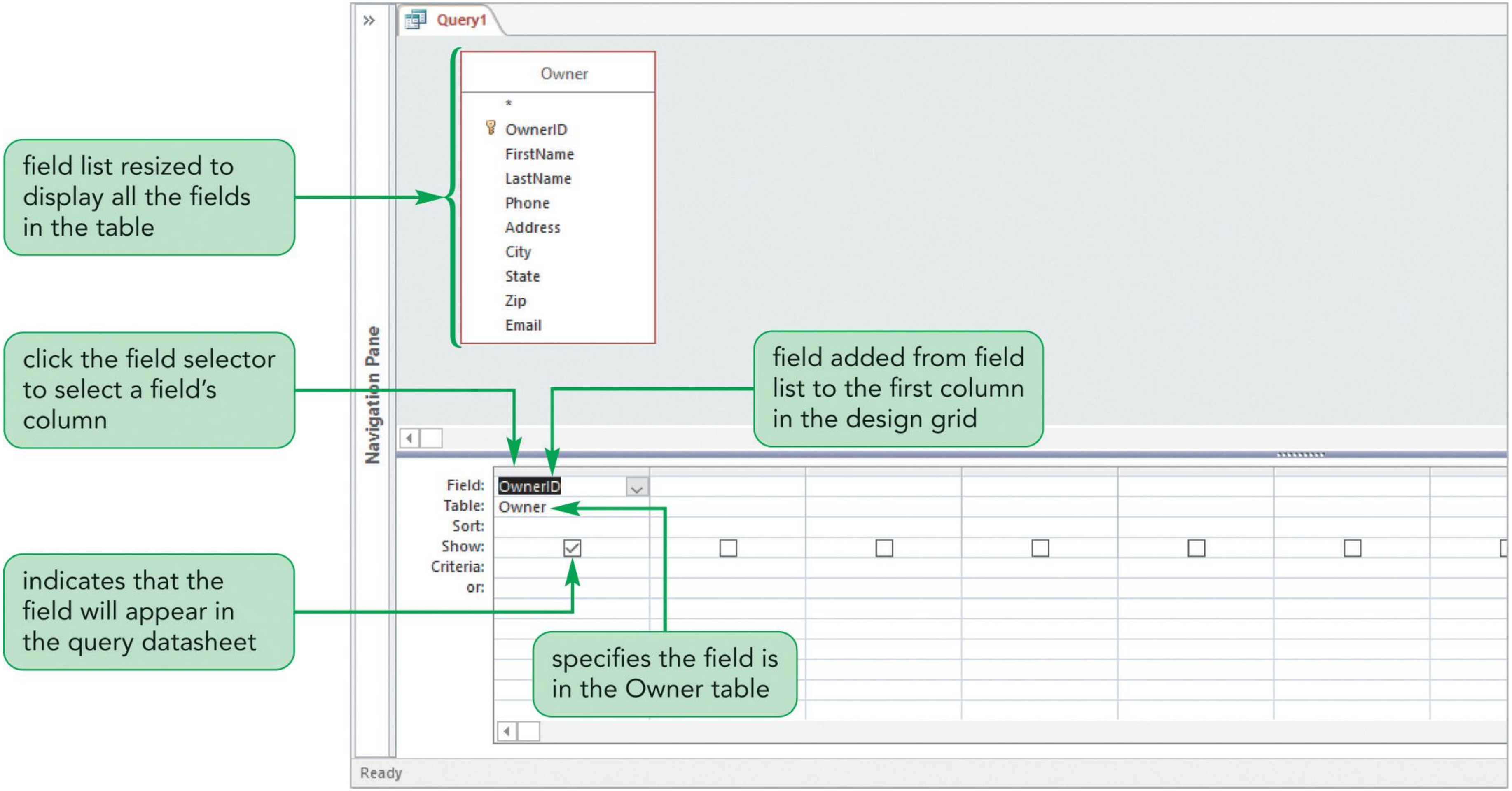
- Create a query based on multiple tables
- Use a comparison operator in a query to match a range of values
- Use the And and Or logical operators in queries
- Create and format a calculated field in a query
- Perform calculations in a query using aggregate functions and record group calculations
- Use 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

GROUP OF EIGHT AUSTRALIA

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



Owner

- *
- OwnerID
- FirstName
- LastName
- Phone
- Address
- City
- State
- Zip
- Email

Field:	OwnerID					
Table:	Owner					
Sort:						
Show:	☑	☐	☐	☐	☐	☐
Criteria:						
or:						

Ready

field list resized to display all the fields in the table

click the field selector to select a field's column

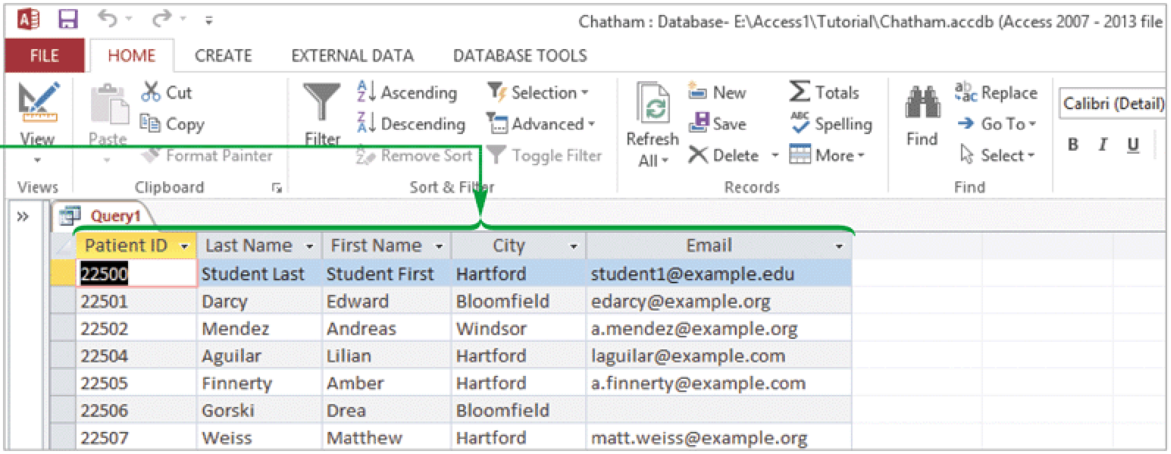
indicates that the field will appear in the query datasheet

field added from field list to the first column in the design grid

specifies the field is in the Owner table

Creating and Running a Query (Cont.)

Figure 3-8 Datasheet displayed after running the query



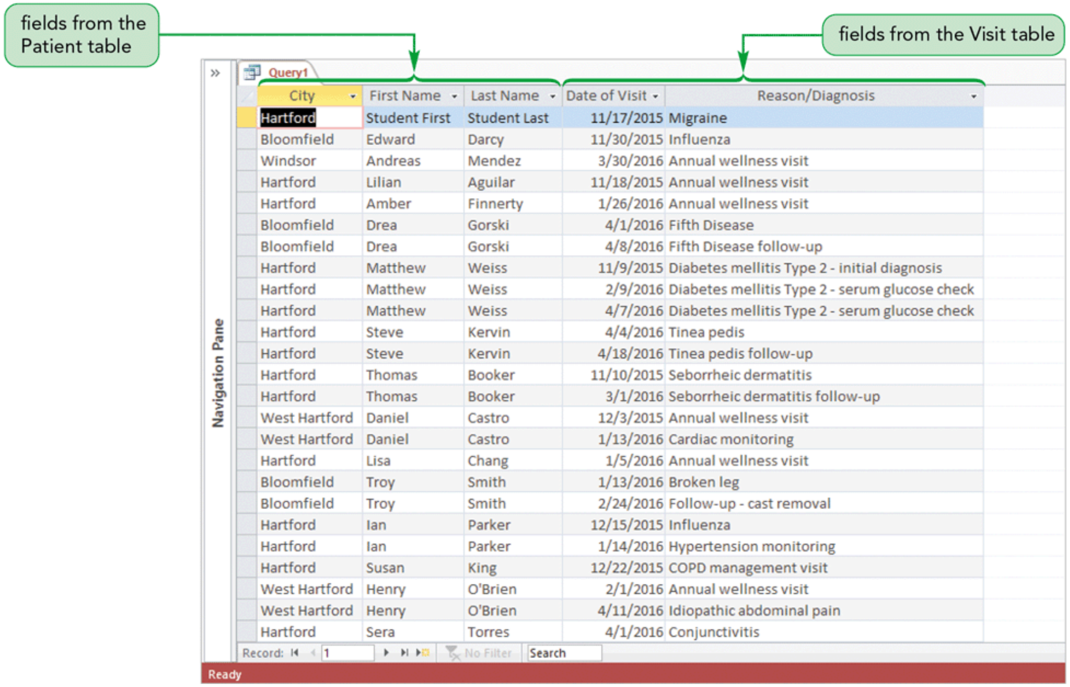
Patient ID	Last Name	First Name	City	Email
22500	Student Last	Student First	Hartford	student1@example.edu
22501	Darcy	Edward	Bloomfield	edarcy@example.org
22502	Mendez	Andreas	Windsor	a.mendez@example.org
22504	Aguilar	Lilian	Hartford	laguilar@example.com
22505	Finnerty	Amber	Hartford	a.finnerty@example.com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

selected fields displayed

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



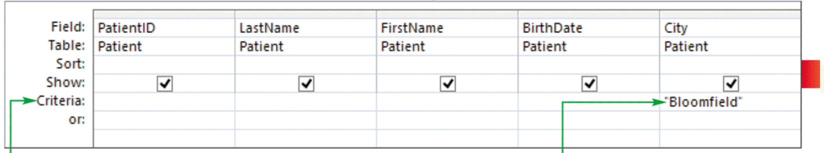
fields from the Patient table

fields from the Visit table

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 mellitis Type 2 - initial diagnosis
Hartford	Matthew	Weiss	2/9/2016	Diabetes mellitis Type 2 - serum glucose check
Hartford	Matthew	Weiss	4/7/2016	Diabetes mellitis 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.

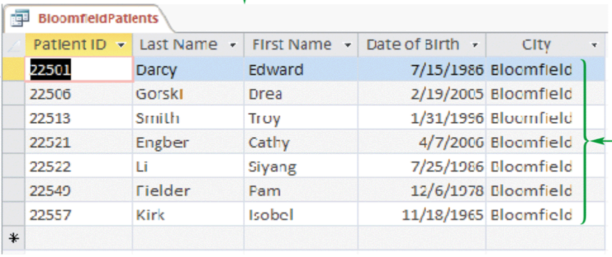


Field:	PatientID	LastName	FirstName	BirthDate	City
Table:	Patient	Patient	Patient	Patient	Patient
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					"Bloomfield"
or:					

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.

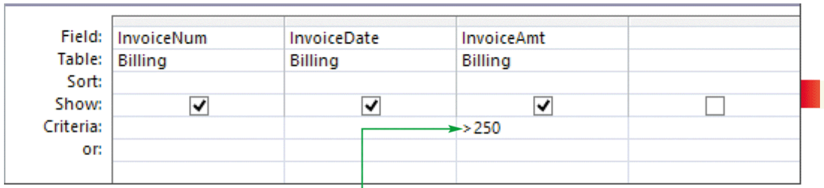
The results of a query containing selection criteria include only the records that meet the specified criteria.



BloomfieldPatients

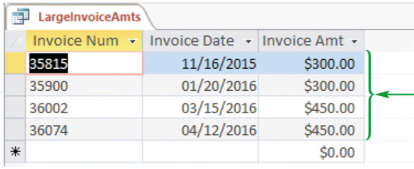
Patient ID	Last Name	First Name	Date of Birth	City
22501	Darcy	Edward	7/15/1986	Bloomfield
22506	Gorski	Drea	2/19/2005	Bloomfield
22513	Smith	Troy	1/31/1996	Bloomfield
22521	Engber	Cathy	4/7/2006	Bloomfield
22522	Li	Siyang	7/25/1986	Bloomfield
22549	Fielder	Pam	12/6/1978	Bloomfield
22557	Kirk	Isobel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



Field:	InvoiceNum	InvoiceDate	InvoiceAmt	
Table:	Billing	Billing	Billing	
Sort:				
Show:	☑	☑	☑	□
Criteria:			>250	
or:				

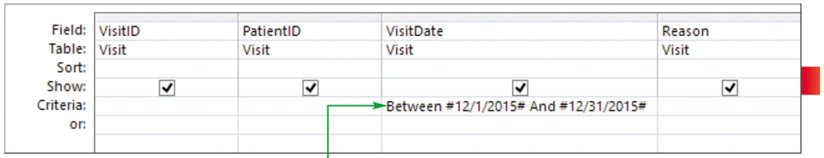
A condition usually consists of an operator, often a comparison operator, and a value. A **comparison operator** asks Access to compare the value in a field to the condition value and to select all the records for which the condition is true.



LargeInvoiceAmts

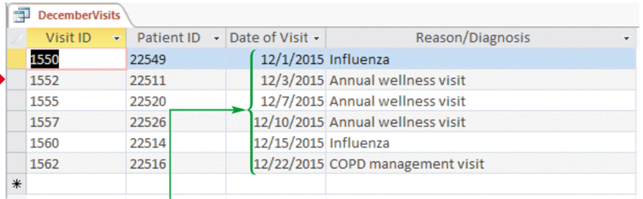
Invoice Num	Invoice Date	Invoice Amt
35815	11/16/2015	$300.00
35900	01/20/2016	$300.00
36002	03/15/2016	$450.00
36074	04/12/2016	$450.00
*		$0.00

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.



Field:	VisitID	PatientID	VisitDate	Reason
Table:	Visit	Visit	Visit	Visit
Sort:				
Show:	☑	☑	☑	☑
Criteria:			Between #12/1/2015# And #12/31/2015#	
or:				

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.



DecemberVisits

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

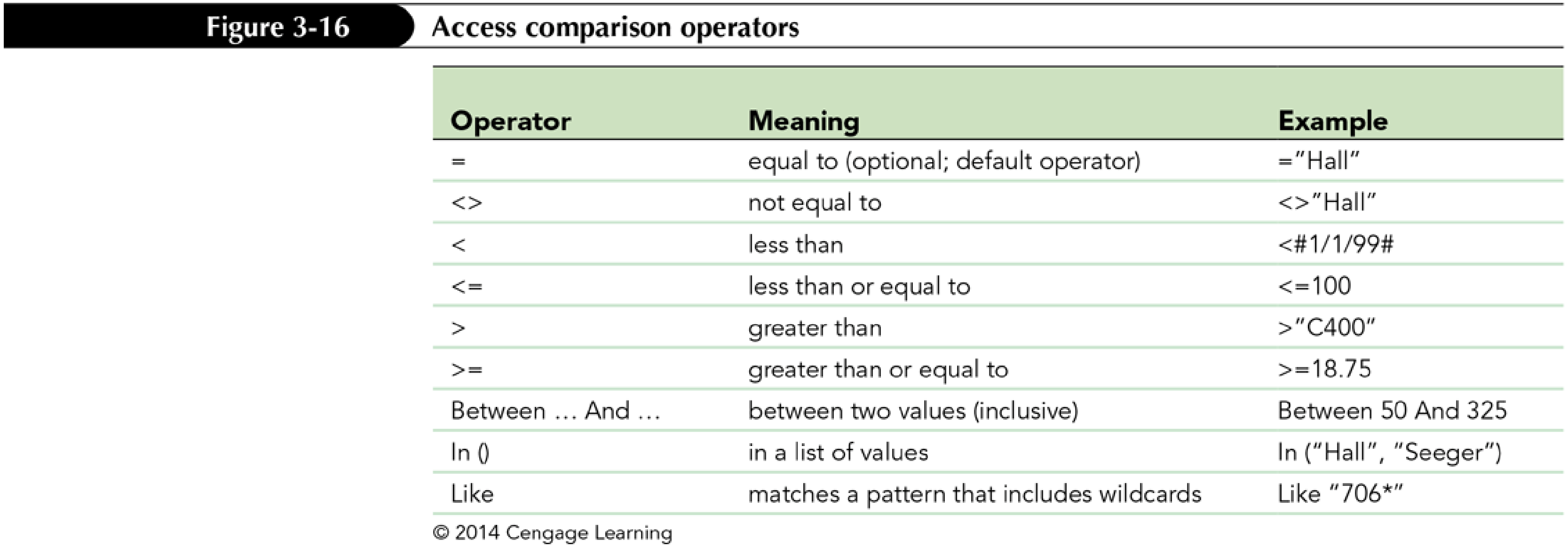
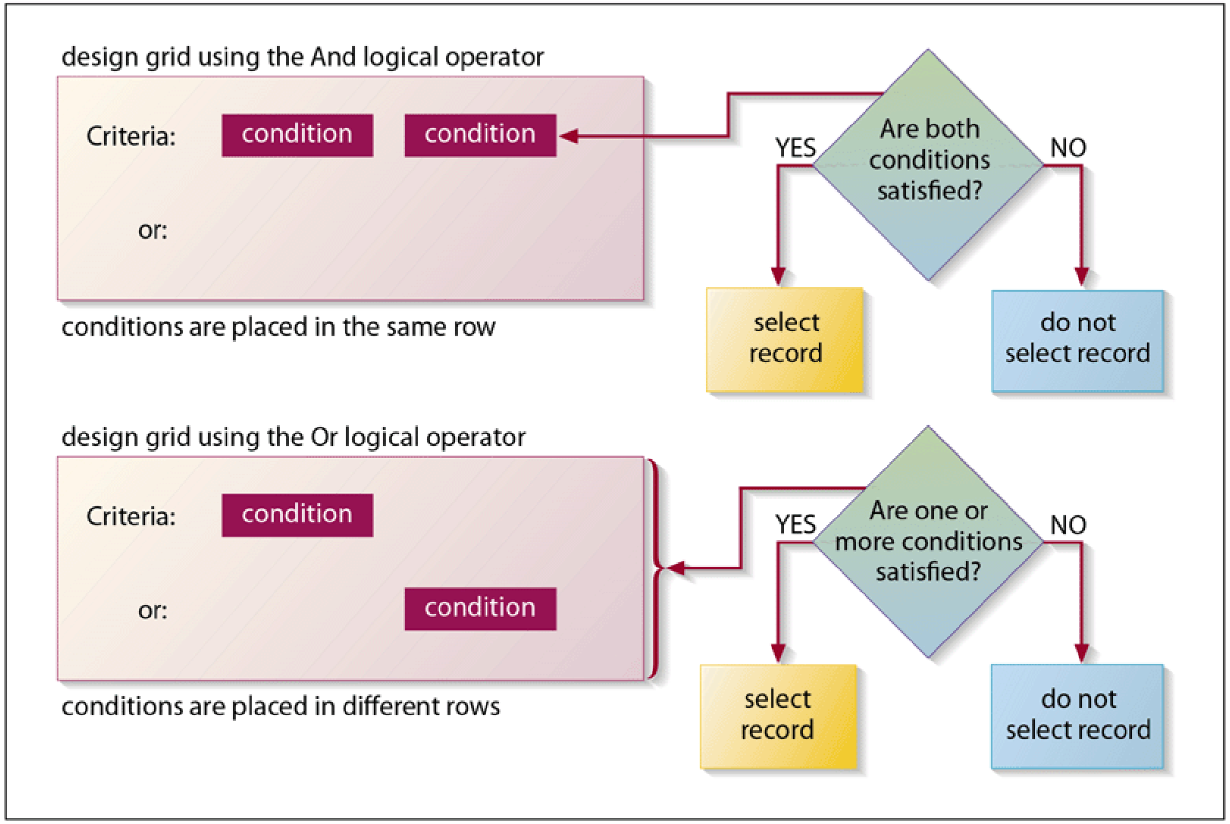


Figure 3-16 Access comparison operators

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*"

Defining Multiple Selection Criteria for Queries (Cont.)

Figure 3-26 Logical operators And and Or for multiple selection criteria


design grid using the And logical operator
Criteria:
condition
condition
or:
conditions are placed in the same row
YES
Are both conditions satisfied?
NO
select record
do not select record
design grid using the Or logical operator
Criteria:
condition
or:
condition
conditions are placed in different rows
YES
Are one or more conditions satisfied?
NO
select record
do not select record

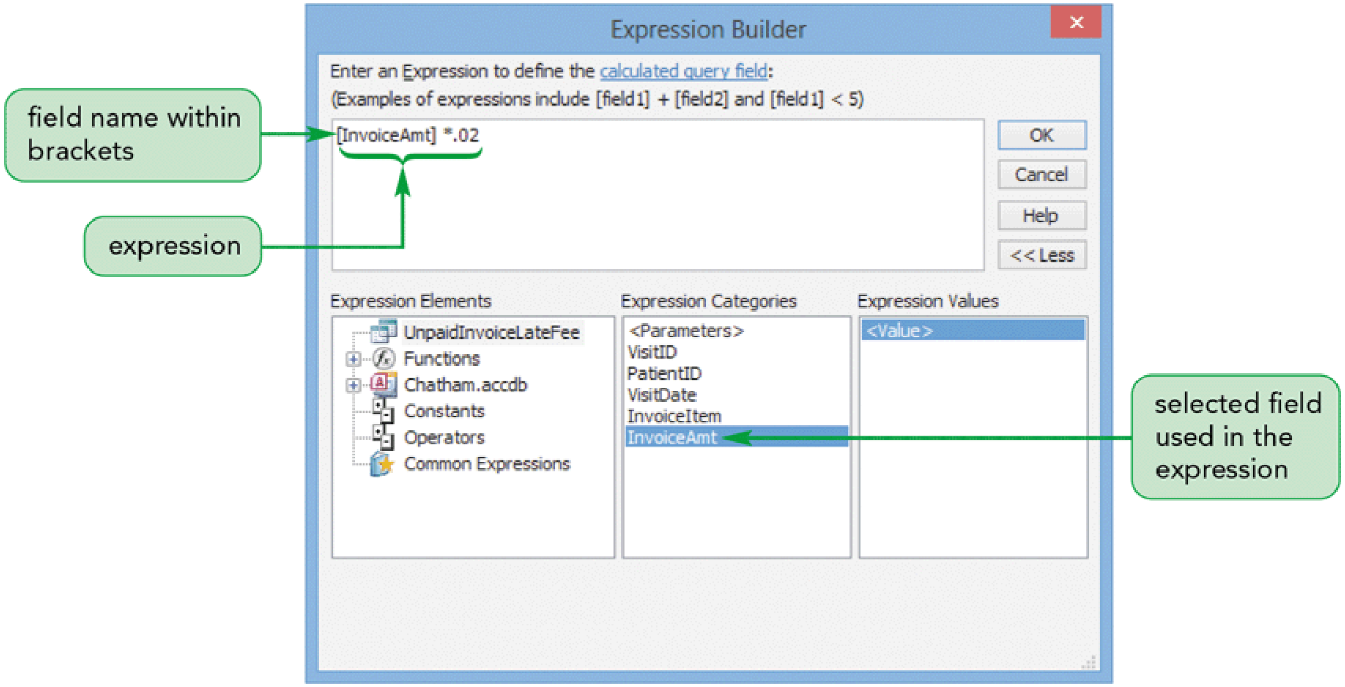
© 2014 Cengage Learning

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



Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

[InvoiceAmt] *.02

field name within brackets

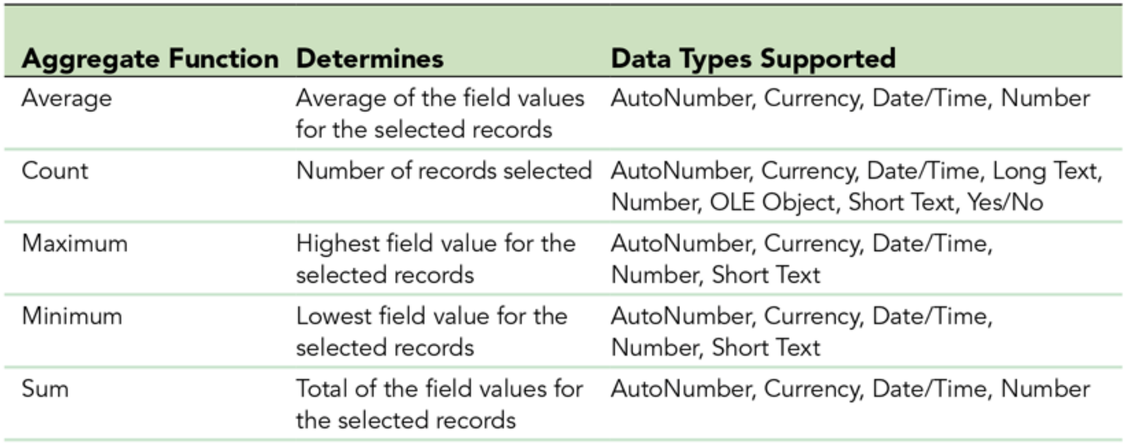
expression

selected field used in the expression

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

Reviewing the Clinic Database

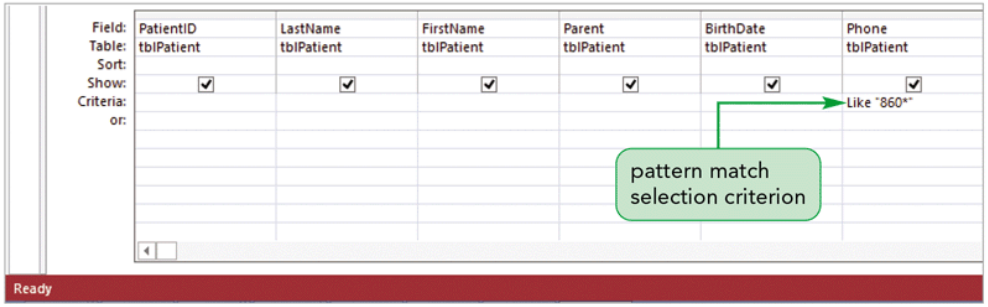
- The Navigation Pane displays the objects grouped by object type
  - Each object name has a prefix tag—a tbl prefix tag for tables, a qry prefix tag for queries, a frm prefix tag for forms, and a rpt prefix tag for reports
  - All three characters in each prefix tag are lower case. The word immediately 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.)

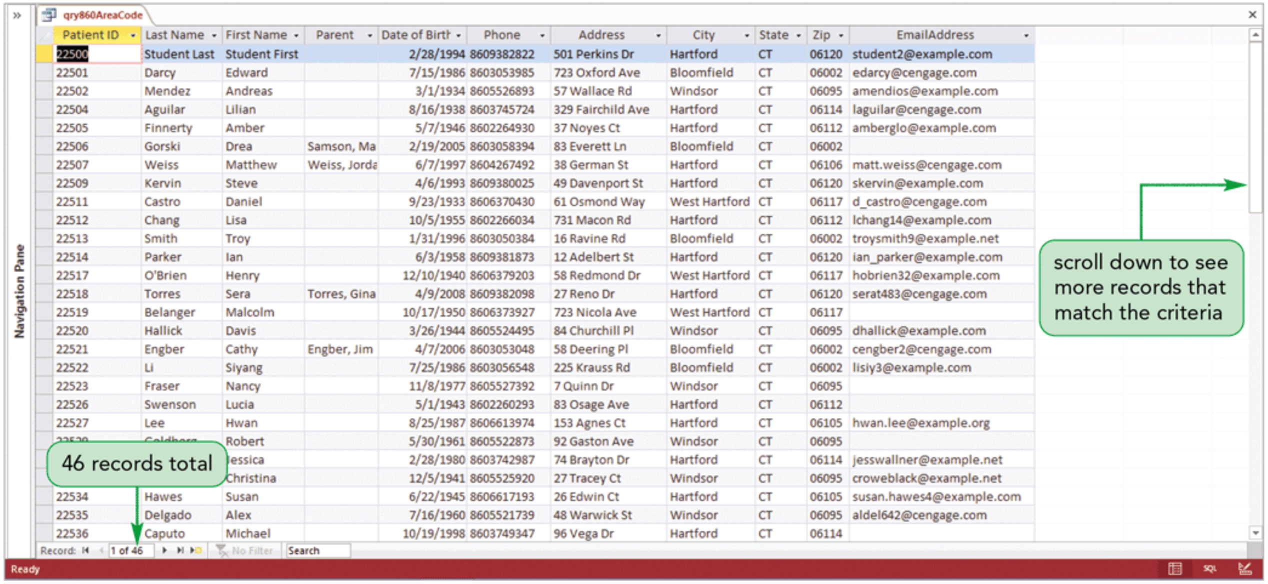
Figure 5-2 Record selection based on matching a specific pattern



Field:	PatientID	LastName	FirstName	Parent	BirthDate	Phone
Table:	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:						Like "860*"
or:						

pattern match selection criterion

Figure 5-3 tblPatient table records for area code 860



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	Lilian		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	amberglo@example.com
22506	Gorski	Drea	Samson, Ma	2/19/2005	8603058394	83 Everett Ln	Bloomfield	CT	06002	
22507	Weiss	Matthew	Weiss, Jorda	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	troysmith9@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
[illegible]	Goldberg	Robert		5/30/1961	8605522873	92 Gaston Ave	Windsor	CT	06095	
[illegible]	[illegible]	Jessica		2/28/1980	8603742987	74 Brayton Dr	Hartford	CT	06114	jesswallner@example.net
[illegible]	[illegible]	Christina		12/5/1941	8605525920	27 Tracey Ct	Windsor	CT	06095	croweblack@example.net
22534	Hawes	Susan		6/22/1945	8606617193	26 Edwin Ct	Hartford	CT	06105	susan.hawes4@example.com
22535	Delgado	Alex		7/16/1960	8605521739	48 Warwick St	Windsor	CT	06095	aldel642@cengage.com
22536	Caputo	Michael		10/19/1998	8603749347	96 Vega Dr	Hartford	CT	06114	

Record: 1 of 46 No Filter Search

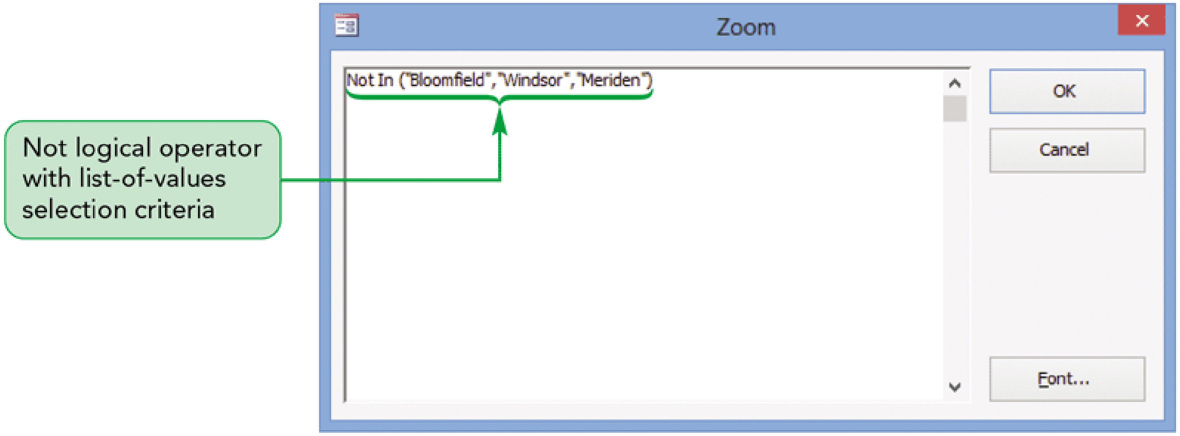
scroll down to see more records that match the criteria

46 records total

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



Not In ("Bloomfield","Windsor","Meriden")

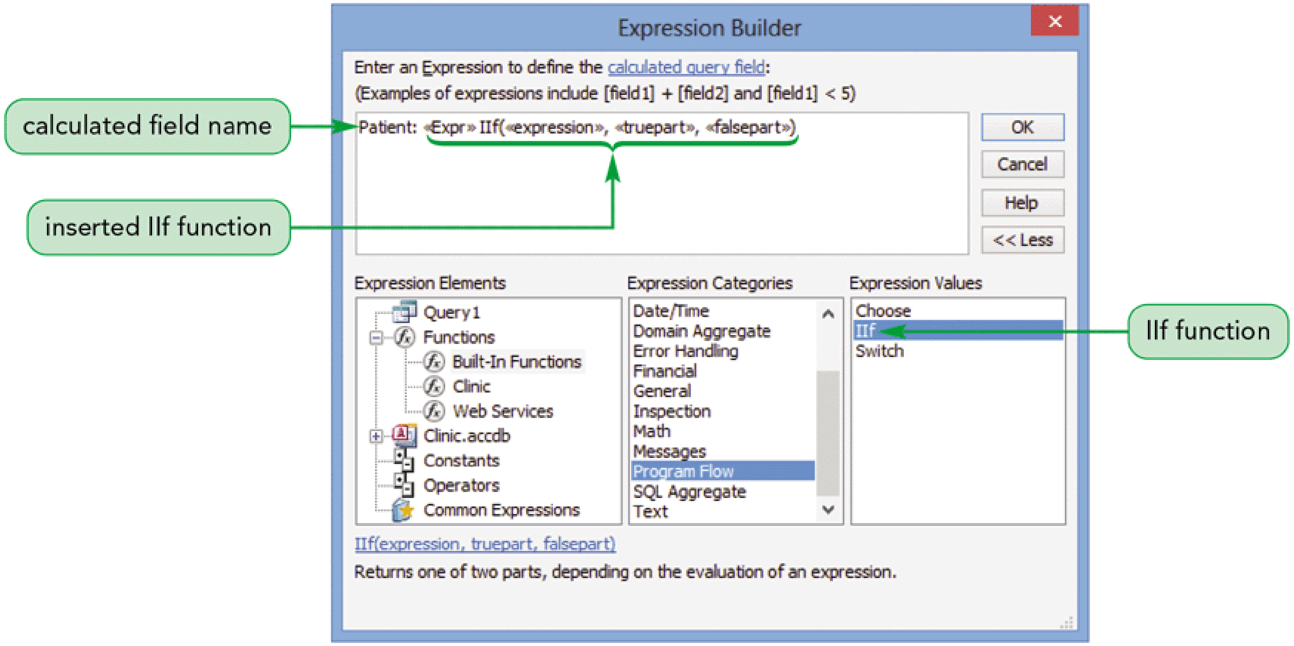
Not logical operator with list-of-values selection criteria

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



calculated field name

inserted IIf function

IIf function

Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

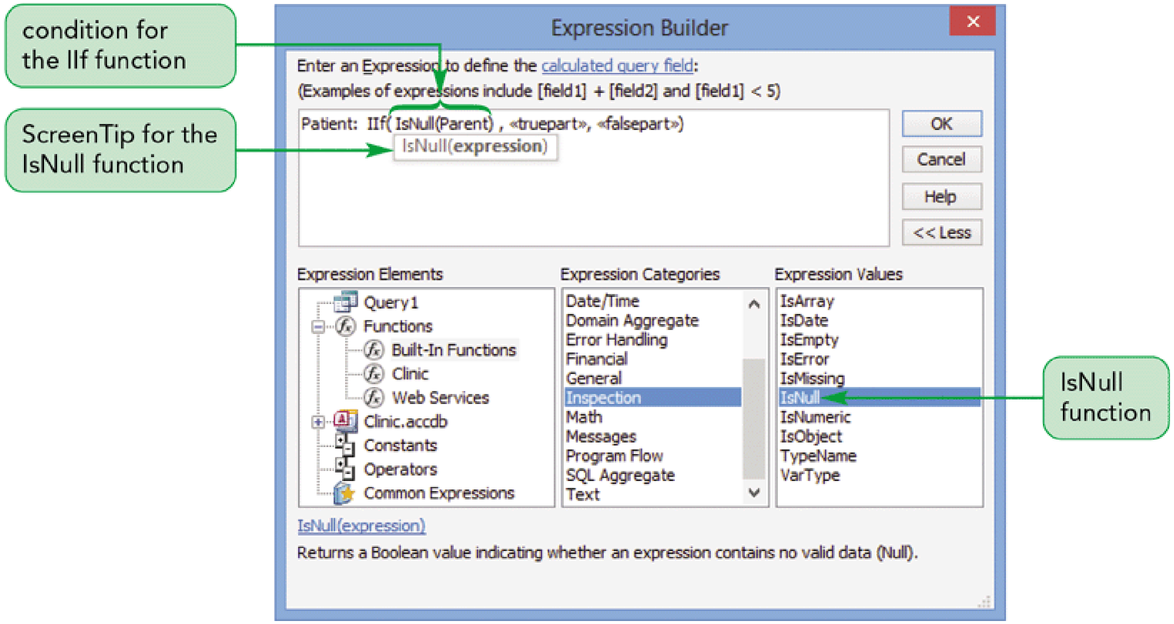
Patient: «Expr» IIf(«expression», «truepart», «falsepart»)

IIf(expression, truepart, falsepart)

Returns one of two parts, depending on the evaluation of an expression.

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

Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

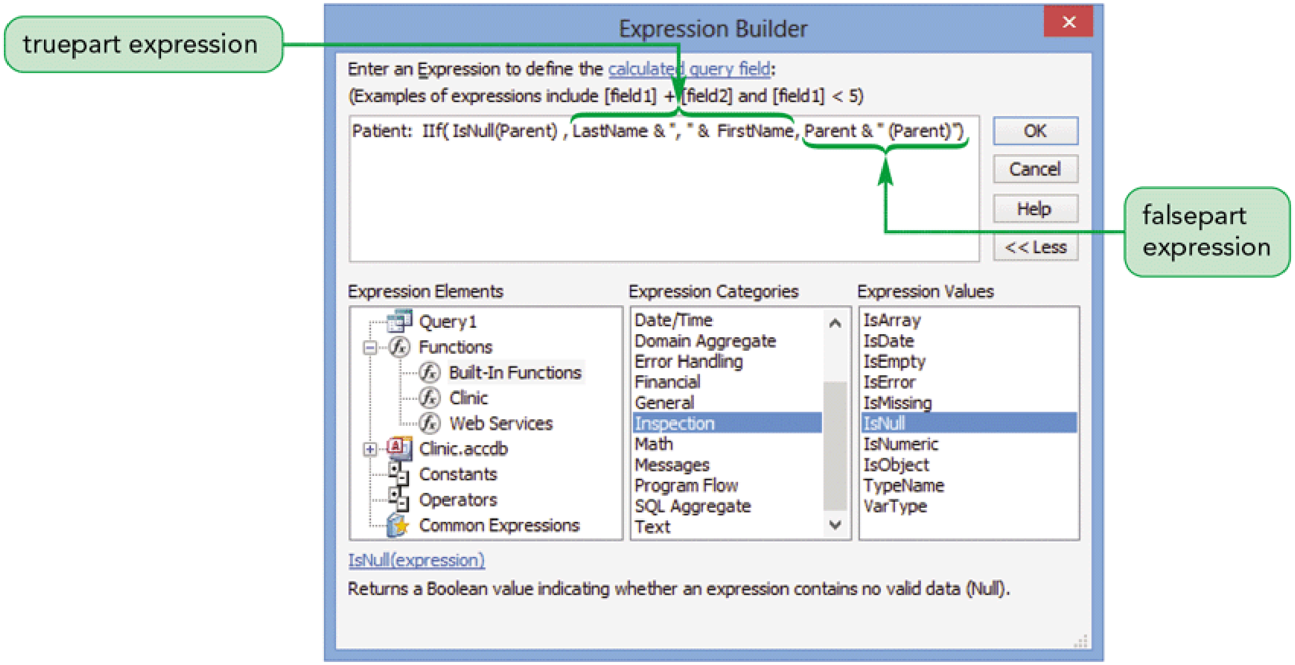
```
Patient: IIf( IsNull(Parent) , «truepart», «falsepart»)
IsNull(expression)
```

IsNull(expression)

Returns a Boolean value indicating whether an expression contains no valid data (Null).

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

Figure 5-10 Completed calculated field



truepart expression

falsepart expression

Expression Builder

Enter an Expression to define the calculated query field:
(Examples of expressions include [field1] + [field2] and [field1] < 5)

```
Patient: IIf( IsNull(Parent) , LastName & ", " & FirstName, Parent & " (Parent)")
```

IsNull(expression)

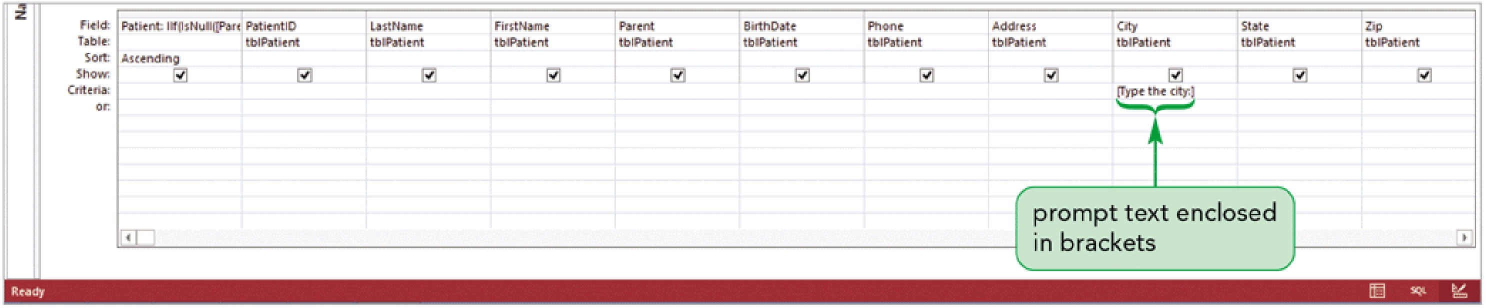
Returns a Boolean value indicating whether an expression contains no valid data (Null).

MONASH University

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



Field:	Patient: IIf(IsNull([Pare	PatientID	LastName	FirstName	Parent	BirthDate	Phone	Address	City	State	Zip
Table:		tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient
Sort:	Ascending										
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Criteria:									[Type the city:]		
or:											

prompt text enclosed in brackets

Creating a Parameter Query (Cont.)

Figure 5-14 Enter Parameter Value dialog box

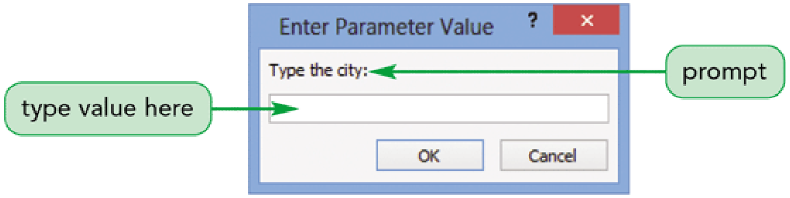
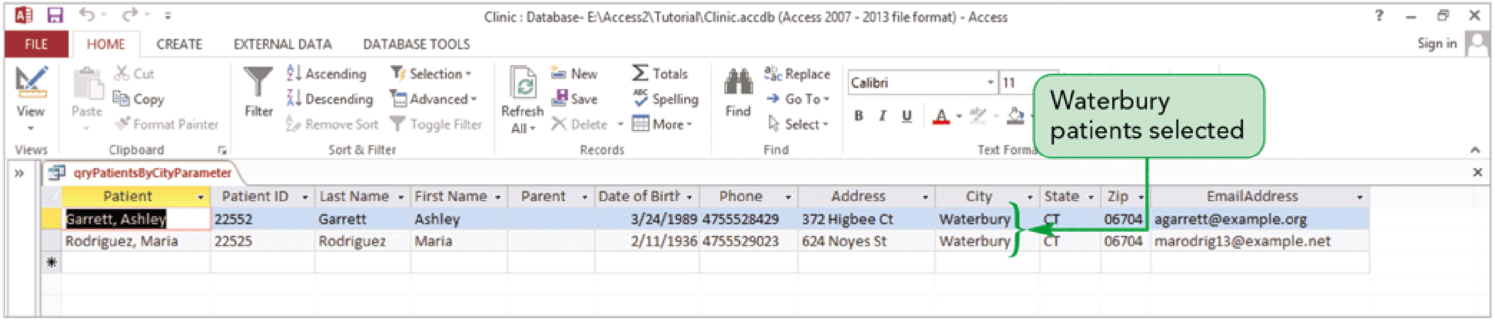


Figure 5-15 Results of the parameter query



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

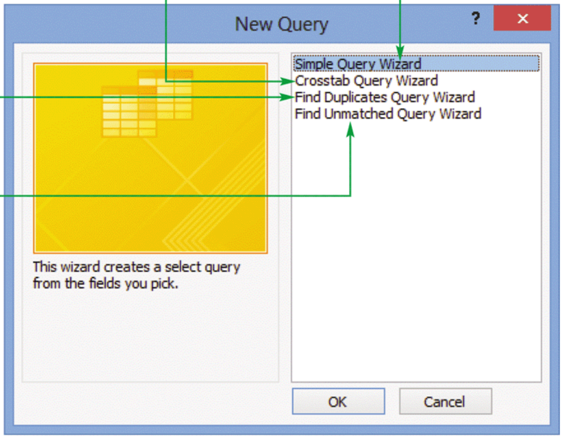
Advanced Query Wizards

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 of more tables that satisfy criteria.

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

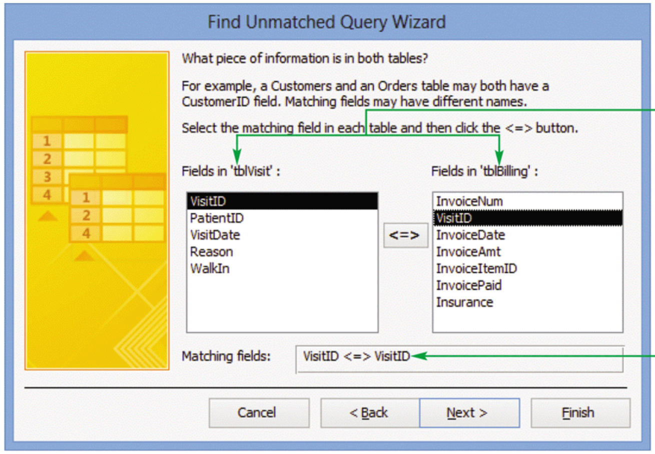
A **find unmatched query** is a select query that finds all records in a table or query that have no related records in a second table or query.



New Query

This wizard creates a select query from the fields you pick.

- Simple Query Wizard
- Crosstab Query Wizard
- Find Duplicates Query Wizard
- Find Unmatched Query Wizard



Find Unmatched Query Wizard

What piece of information is in both tables?

For example, a Customers and an Orders table may both have a CustomerID field. Matching fields may have different names.

Select the matching field in each table and then click the <=> button.

Fields in 'tblVisit' :

- VisitID
- PatientID
- VisitDate
- Reason
- WalkIn

Fields in 'tblBilling' :

- InvoiceNum
- VisitID
- InvoiceDate
- InvoiceAmt
- InvoiceItemID
- InvoicePaid
- Insurance

Matching fields: VisitID <=> VisitID

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.

This list contains the remaining fields in the tblVisit table that will not be considered for duplicate values.

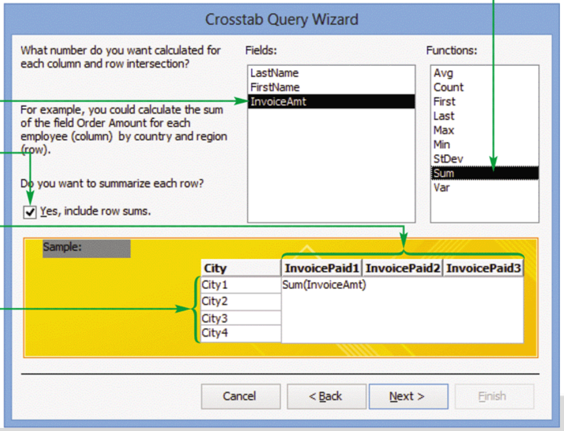
Each column and row intersection will display the sum of the InvoiceAmt values.

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.



Crosstab Query Wizard

What number do you want calculated for each column and row intersection?

For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row).

Do you want to summarize each row?

☑ Yes, include row sums.

Fields:

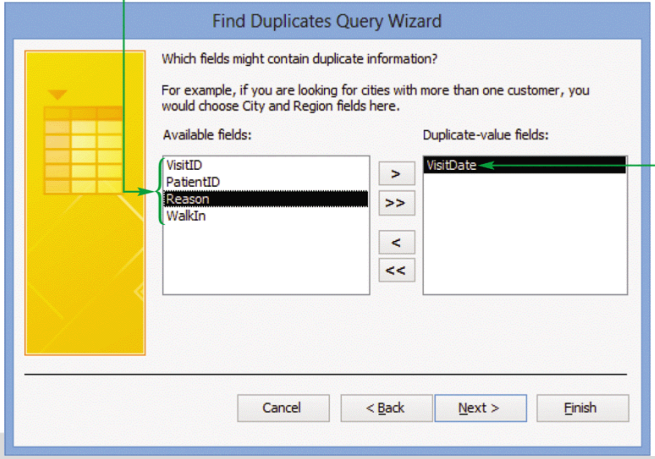
- LastName
- FirstName
- InvoiceAmt

Functions:

- Avg
- Count
- First
- Last
- Max
- Min
- StDev
- Sum
- Var

Sample:

City	InvoicePaid1	InvoicePaid2	InvoicePaid3
City1	Sum(InvoiceAmt)		
City2			
City3			
City4			



Find Duplicates Query Wizard

Which fields might contain duplicate information?

For example, if you are looking for cities with more than one customer, you would choose City and Region fields here.

Available fields:

- VisitID
- PatientID
- Reason
- WalkIn

Duplicate-value fields:

- VisitDate

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

Creating a Crosstab Query

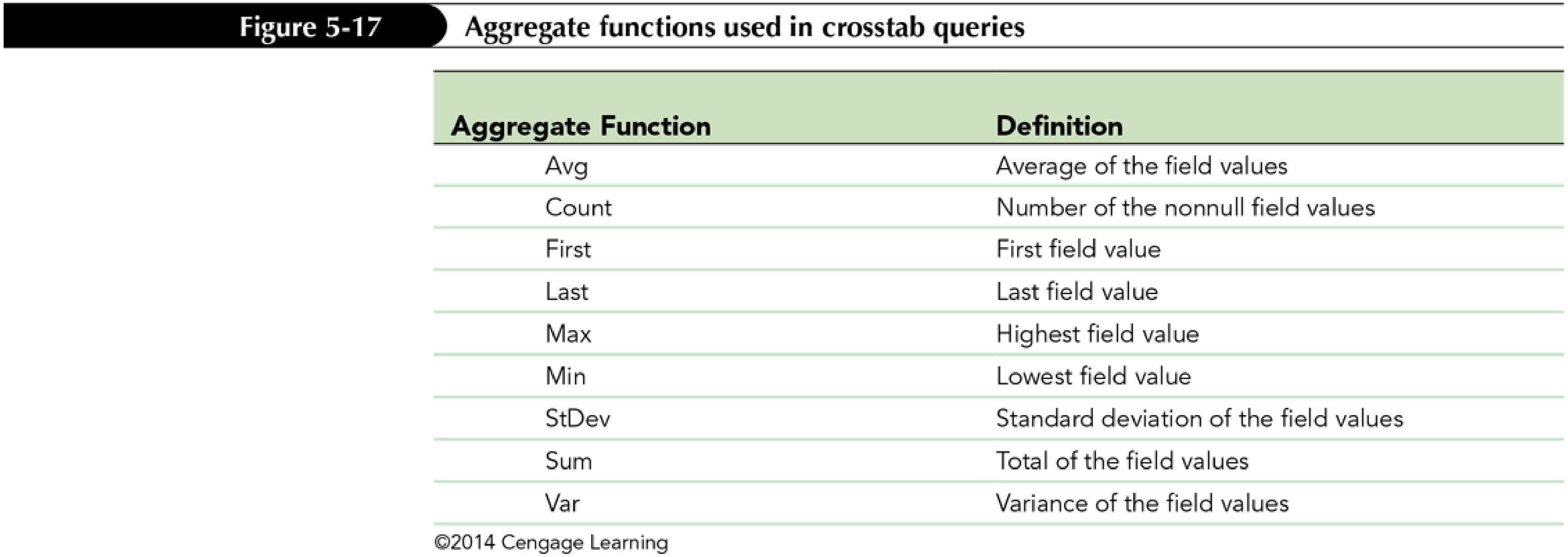
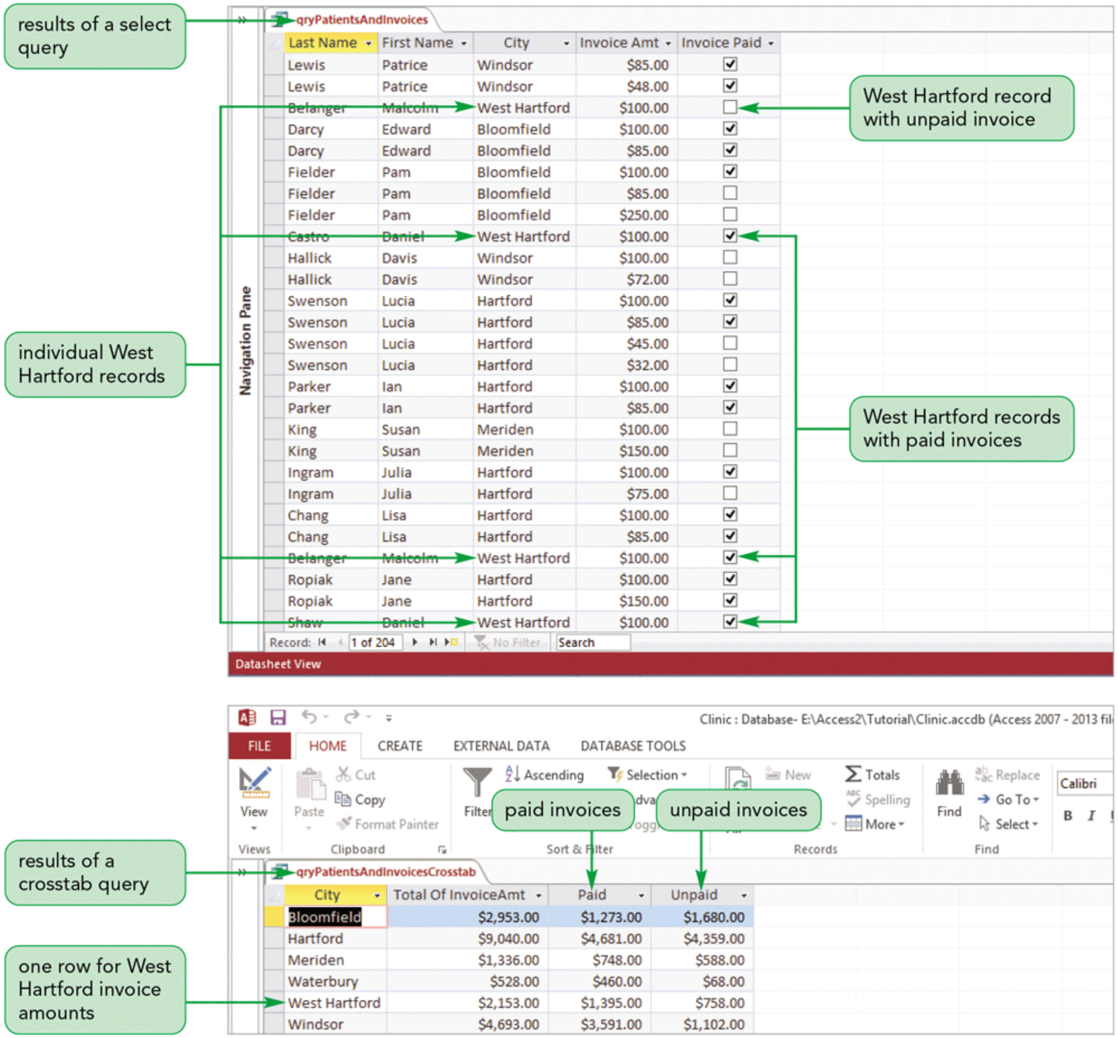


Figure 5-17 Aggregate functions used in crosstab queries

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

Creating a Crosstab Query (Cont.)

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



results of a select query
West Hartford record with unpaid invoice
individual West Hartford records
West Hartford records with paid invoices
Navigation Pane
Record: 1 of 204
No Filter
Search
Datasheet View
Clinic : Database- E:\Access2\Tutorial\Clinic.accdb (Access 2007 - 2013 fil
FILE
HOME
CREATE
EXTERNAL DATA
DATABASE TOOLS
View
Paste
Cut
Copy
Format Painter
Filter
Ascending
Selection
New
Totals
Spelling
More
Find
Replace
Go To
Select
Calibri
Views
Clipboard
Sort & Filter
Records
Find
paid invoices
unpaid invoices
results of a crosstab query
one row for West Hartford invoice amounts

qryPatientsAndInvoices

Last Name	First Name	City	Invoice Amt	Invoice Paid
Lewis	Patrice	Windsor	$85.00	☑
Lewis	Patrice	Windsor	$48.00	☑
Belanger	Malcolm	West Hartford	$100.00	□
Darcy	Edward	Bloomfield	$100.00	☑
Darcy	Edward	Bloomfield	$85.00	☑
Fielder	Pam	Bloomfield	$100.00	☑
Fielder	Pam	Bloomfield	$85.00	□
Fielder	Pam	Bloomfield	$250.00	□
Castro	Daniel	West Hartford	$100.00	☑
Hallick	Davis	Windsor	$100.00	□
Hallick	Davis	Windsor	$72.00	□
Swenson	Lucia	Hartford	$100.00	☑
Swenson	Lucia	Hartford	$85.00	☑
Swenson	Lucia	Hartford	$45.00	□
Swenson	Lucia	Hartford	$32.00	□
Parker	Ian	Hartford	$100.00	☑
Parker	Ian	Hartford	$85.00	☑
King	Susan	Meriden	$100.00	□
King	Susan	Meriden	$150.00	□
Ingram	Julia	Hartford	$100.00	☑
Ingram	Julia	Hartford	$75.00	□
Chang	Lisa	Hartford	$100.00	☑
Chang	Lisa	Hartford	$85.00	☑
Belanger	Malcolm	West Hartford	$100.00	☑
Ropiak	Jane	Hartford	$100.00	☑
Ropiak	Jane	Hartford	$150.00	☑
Shaw	Daniel	West Hartford	$100.00	☑

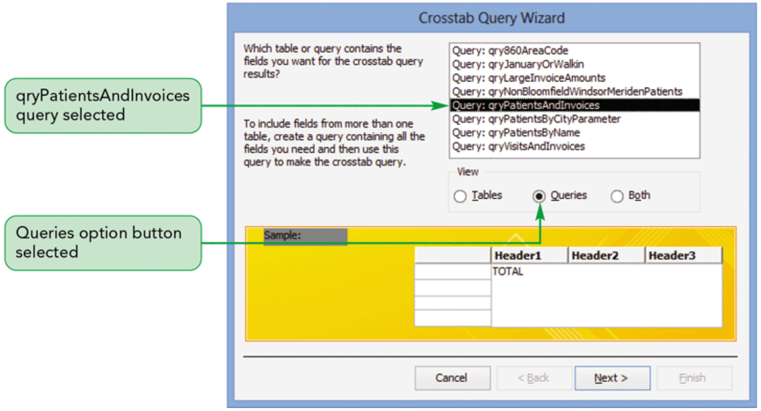
qryPatientsAndInvoicesCrosstab

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



Which table or query contains the fields you want for the crosstab query results?

Query: qry860AreaCode
Query: qryJanuaryOrWalkin
Query: qryLargeInvoiceAmounts
Query: qryNonBloomfieldWindsorMeridenPatients
Query: qryPatientsAndInvoices
Query: qryPatientsByCityParameter
Query: qryPatientsByName
Query: qryVisitsAndInvoices

To include fields from more than one table, create a query containing all the fields you need and then use this query to make the crosstab query.

View
○ Tables ● Queries ○ Both

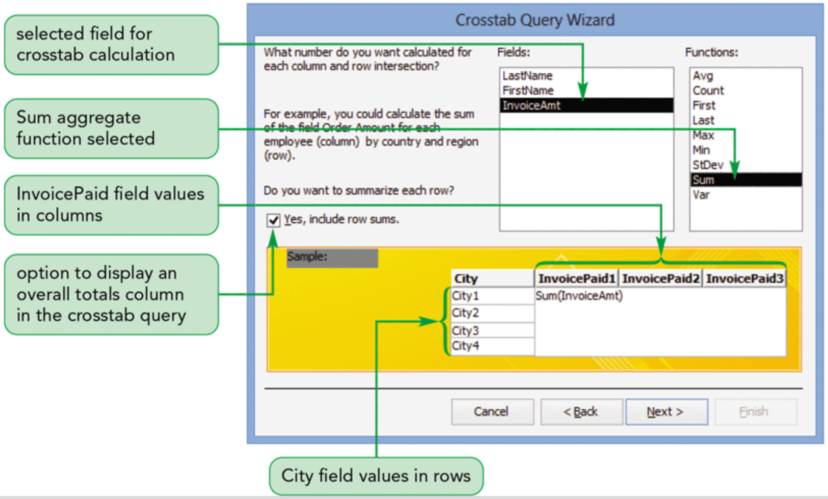
Sample:

	Header1	Header2	Header3
	TOTAL		

qryPatientsAndInvoices query selected

Queries option button selected

Figure 5-20 Completed crosstab query design



What number do you want calculated for each column and row intersection?

Fields:
LastName
FirstName
InvoiceAmt

Functions:
Avg
Count
First
Last
Max
Min
StDev
Sum
Var

For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row).

Do you want to summarize each row?

☑ Yes, include row sums.

Sample:

City	InvoicePaid1	InvoicePaid2	InvoicePaid3
City1	Sum(InvoiceAmt)		
City2			
City3			
City4			

selected field for crosstab calculation

Sum aggregate function selected

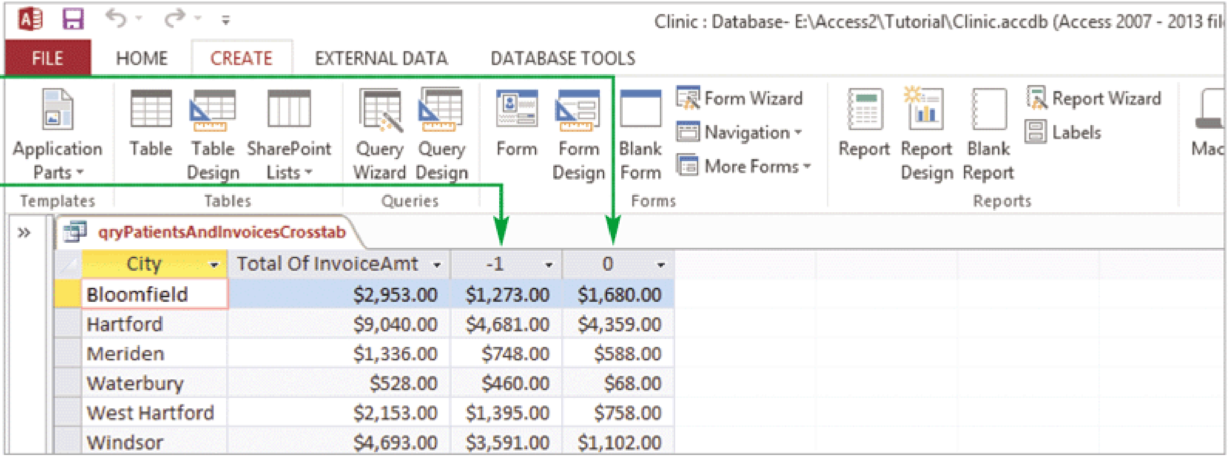
InvoicePaid field values in columns

option to display an overall totals column in the crosstab query

City field values in rows

Creating a Crosstab Query (Cont.)

Figure 5-21 Crosstab query recordset

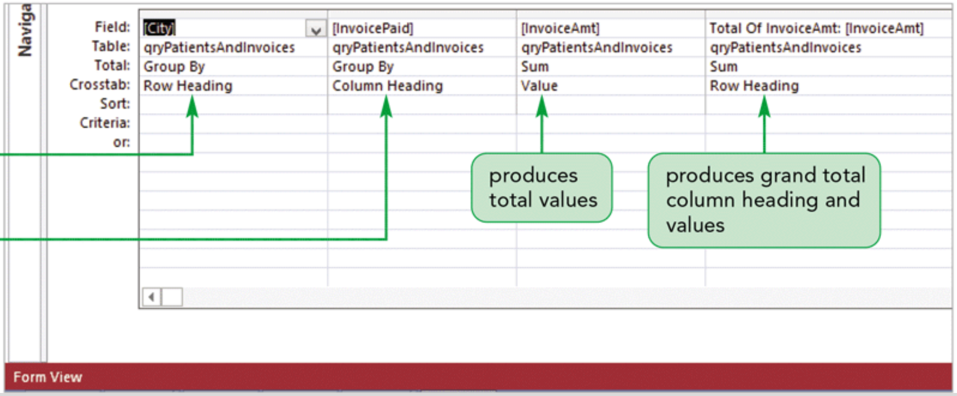


City	Total Of InvoiceAmt	-1	0
Bloomfield	$2,953.00	$1,273.00	$1,680.00
Hartford	$9,040.00	$4,681.00	$4,359.00
Meriden	$1,336.00	$748.00	$588.00
Waterbury	$528.00	$460.00	$68.00
West Hartford	$2,153.00	$1,395.00	$758.00
Windsor	$4,693.00	$3,591.00	$1,102.00

unpaid invoices by city

paid invoices by city

Figure 5-22 Crosstab query in the design grid



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

produces row headings

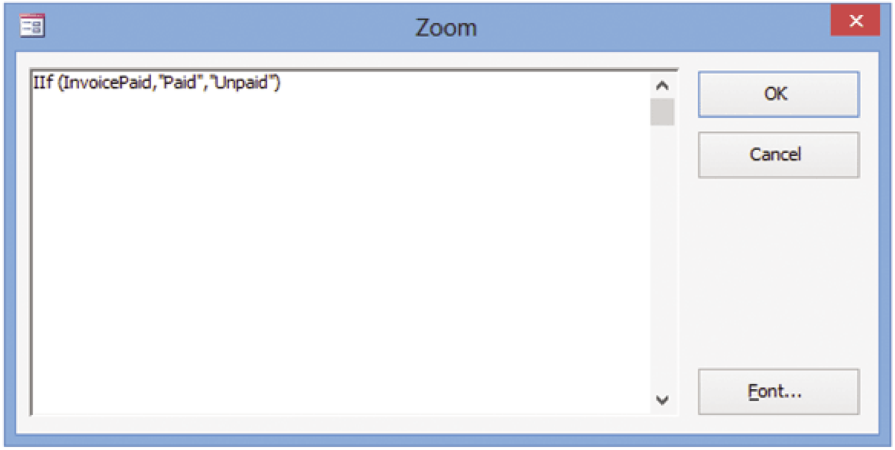
produces column headings

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

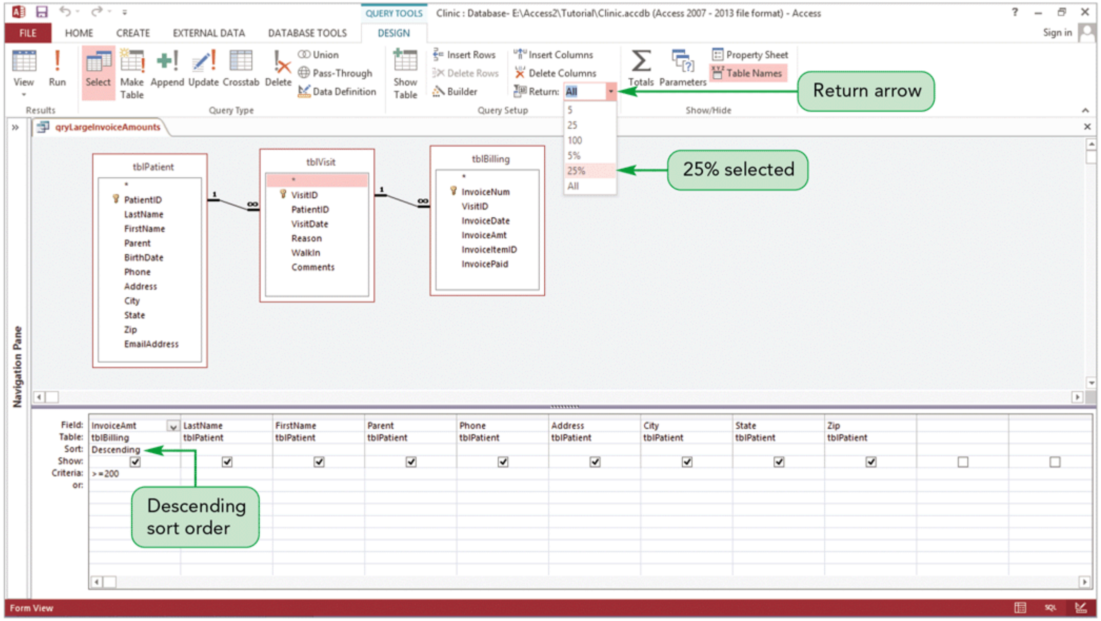


IIf (InvoicePaid,"Paid","Unpaid")

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

Figure 5-27 Creating the top values query



Return: All
5
25
100
5%
25%
All

Return arrow

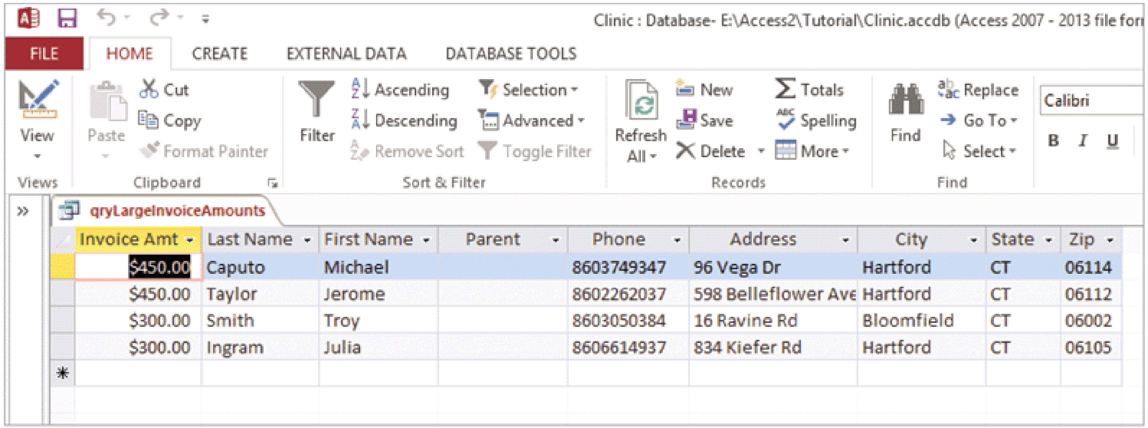
25% selected

Field:	InvoiceAmt	LastName	FirstName	Parent	Phone	Address	City	State	Zip		
Table:	tblBilling	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient	tblPatient		
Sort:	Descending										
Show:	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☐
Criteria:	>=200										
or:											

Descending sort order

Creating a Top Values Query (Cont.)

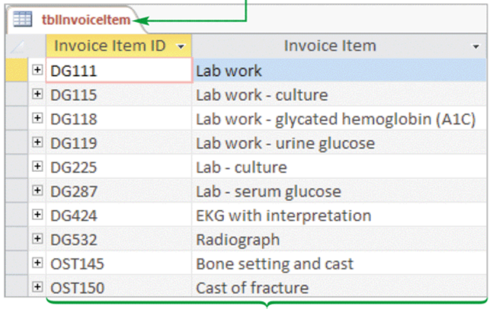
Figure 5-28 Top values query recordset



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.

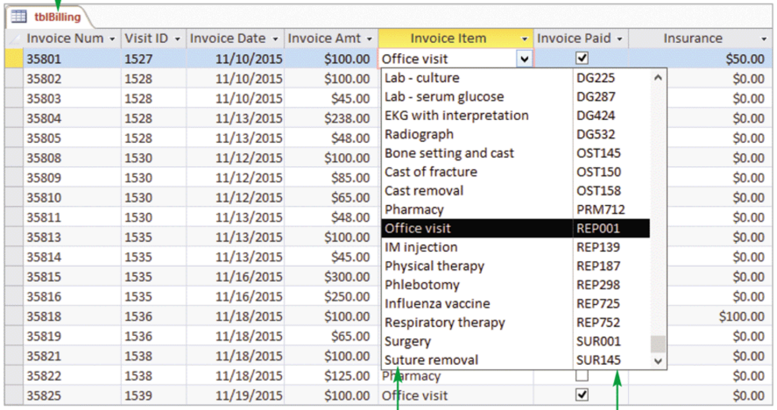


tblInvoiceItem

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

The tblBilling table contains the lookup field.



tblBilling

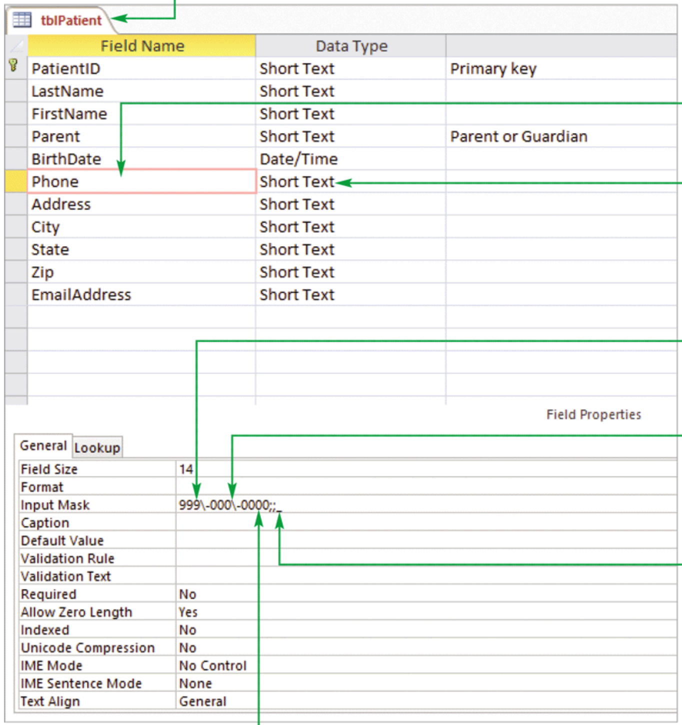
Invoice Num	Visit ID	Invoice Date	Invoice Amt	Invoice Item	Invoice Paid	Insurance
35801	1527	11/10/2015	$100.00	Office visit	☑	$50.00
35802	1528	11/10/2015	$100.00			$0.00
35803	1528	11/10/2015	$45.00			$0.00
35804	1528	11/13/2015	$238.00			$0.00
35805	1528	11/13/2015	$48.00			$0.00
35808	1530	11/12/2015	$100.00			$0.00
35809	1530	11/12/2015	$85.00			$0.00
35810	1530	11/12/2015	$65.00			$0.00
35811	1530	11/13/2015	$48.00			$0.00
35813	1535	11/13/2015	$100.00			$0.00
35814	1535	11/13/2015	$45.00			$0.00
35815	1535	11/16/2015	$300.00			$0.00
35816	1535	11/16/2015	$250.00			$0.00
35818	1536	11/18/2015	$100.00			$100.00
35819	1536	11/18/2015	$65.00			$0.00
35821	1538	11/18/2015	$100.00			$0.00
35822	1538	11/18/2015	$125.00	Pharmacy	□	$0.00
35825	1539	11/19/2015	$100.00	Office visit	☑	$0.00

Lab - culture	DG225
Lab - serum glucose	DG287
EKG with interpretation	DG424
Radiograph	DG532
Bone setting and cast	OST145
Cast of fracture	OST150
Cast removal	OST158
Pharmacy	PRM712
Office visit	REP001
IM injection	REP139
Physical therapy	REP187
Phlebotomy	REP298
Influenza vaccine	REP725
Respiratory therapy	REP752
Surgery	SUR001
Suture removal	SUR145

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.



tblPatient

Field Name	Data Type	
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	

Field Properties

General | Lookup

Field Size	14
Format	
Input Mask	999\-000\-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

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.

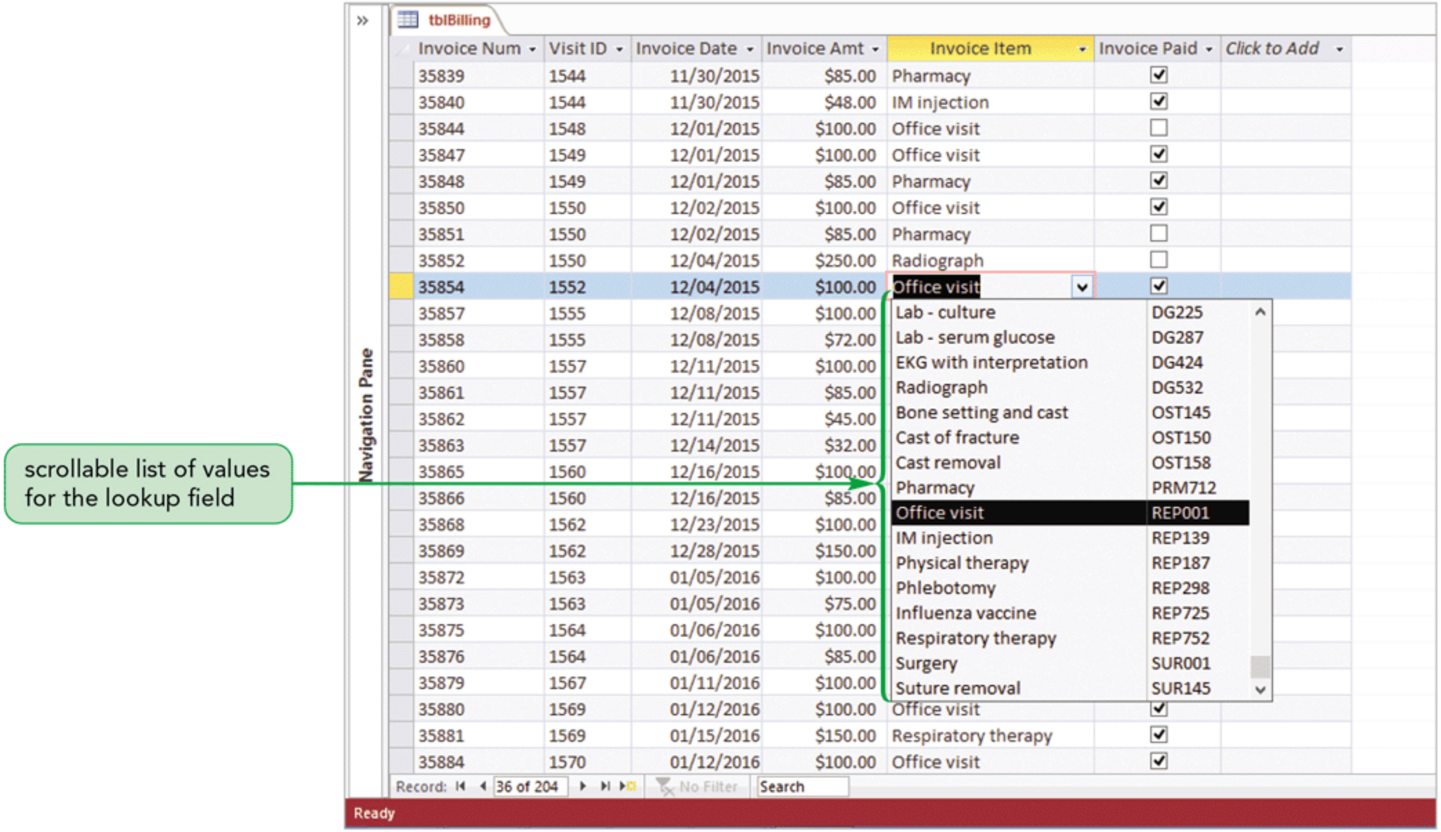
The \ indicates that the character that follows is a literal display character.

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.

Creating a Lookup Field (Cont.)

Figure 5-32 List of InvoiceItem and InvoiceItemID field values



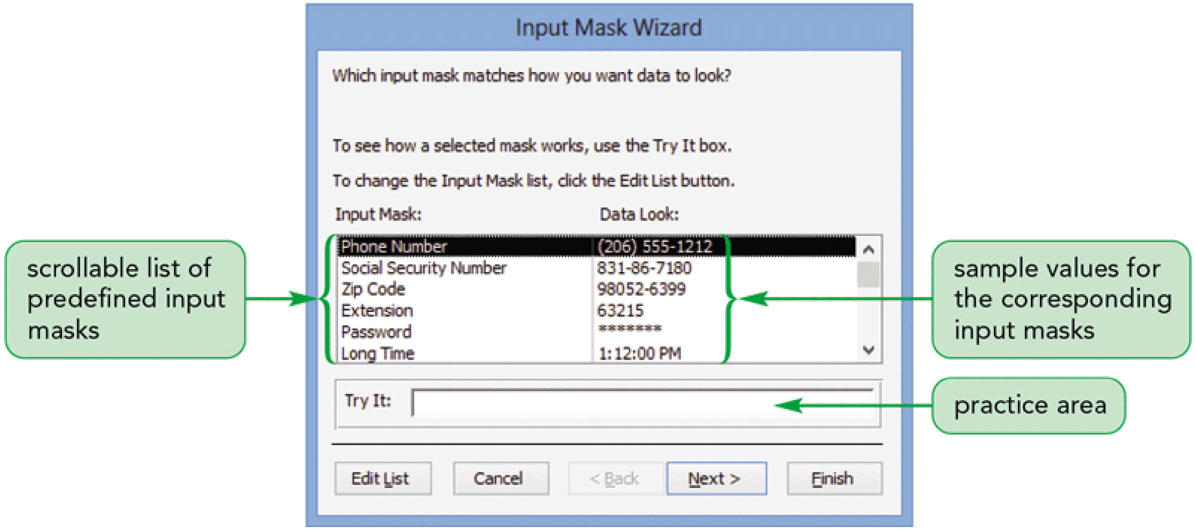
scrollable list of values for the lookup field

Invoice Num	Visit ID	Invoice Date	Invoice Amt	Invoice Item	Invoice Paid	Click to Add
35839	1544	11/30/2015	$85.00	Pharmacy	☑	
35840	1544	11/30/2015	$48.00	IM injection	☑	
35844	1548	12/01/2015	$100.00	Office visit	□	
35847	1549	12/01/2015	$100.00	Office visit	☑	
35848	1549	12/01/2015	$85.00	Pharmacy	☑	
35850	1550	12/02/2015	$100.00	Office visit	☑	
35851	1550	12/02/2015	$85.00	Pharmacy	□	
35852	1550	12/04/2015	$250.00	Radiograph	□	
35854	1552	12/04/2015	$100.00	Office visit	☑	
35857	1555	12/08/2015	$100.00			
35858	1555	12/08/2015	$72.00			
35860	1557	12/11/2015	$100.00			
35861	1557	12/11/2015	$85.00			
35862	1557	12/11/2015	$45.00			
35863	1557	12/14/2015	$32.00			
35865	1560	12/16/2015	$100.00			
35866	1560	12/16/2015	$85.00			
35868	1562	12/23/2015	$100.00			
35869	1562	12/28/2015	$150.00			
35872	1563	01/05/2016	$100.00			
35873	1563	01/05/2016	$75.00			
35875	1564	01/06/2016	$100.00			
35876	1564	01/06/2016	$85.00			
35879	1567	01/11/2016	$100.00			
35880	1569	01/12/2016	$100.00	Office visit	☑	
35881	1569	01/15/2016	$150.00	Respiratory therapy	☑	
35884	1570	01/12/2016	$100.00	Office visit	☑	

Lab - culture	DG225
Lab - serum glucose	DG287
EKG with interpretation	DG424
Radiograph	DG532
Bone setting and cast	OST145
Cast of fracture	OST150
Cast removal	OST158
Pharmacy	PRM712
Office visit	REP001
IM injection	REP139
Physical therapy	REP187
Phlebotomy	REP298
Influenza vaccine	REP725
Respiratory therapy	REP752
Surgery	SUR001
Suture removal	SUR145

Using the Input Mask Wizard (Cont.)

Figure 5-33 Input Mask Wizard dialog box



Input Mask Wizard

Which input mask matches how you want data to look?

To see how a selected mask works, use the Try It box.

To change the Input Mask list, click the Edit List button.

Input Mask:	Data Look:
Phone Number	(206) 555-1212
Social Security Number	831-86-7180
Zip Code	98052-6399
Extension	63215
Password	********
Long Time	1:12:00 PM

Try It:

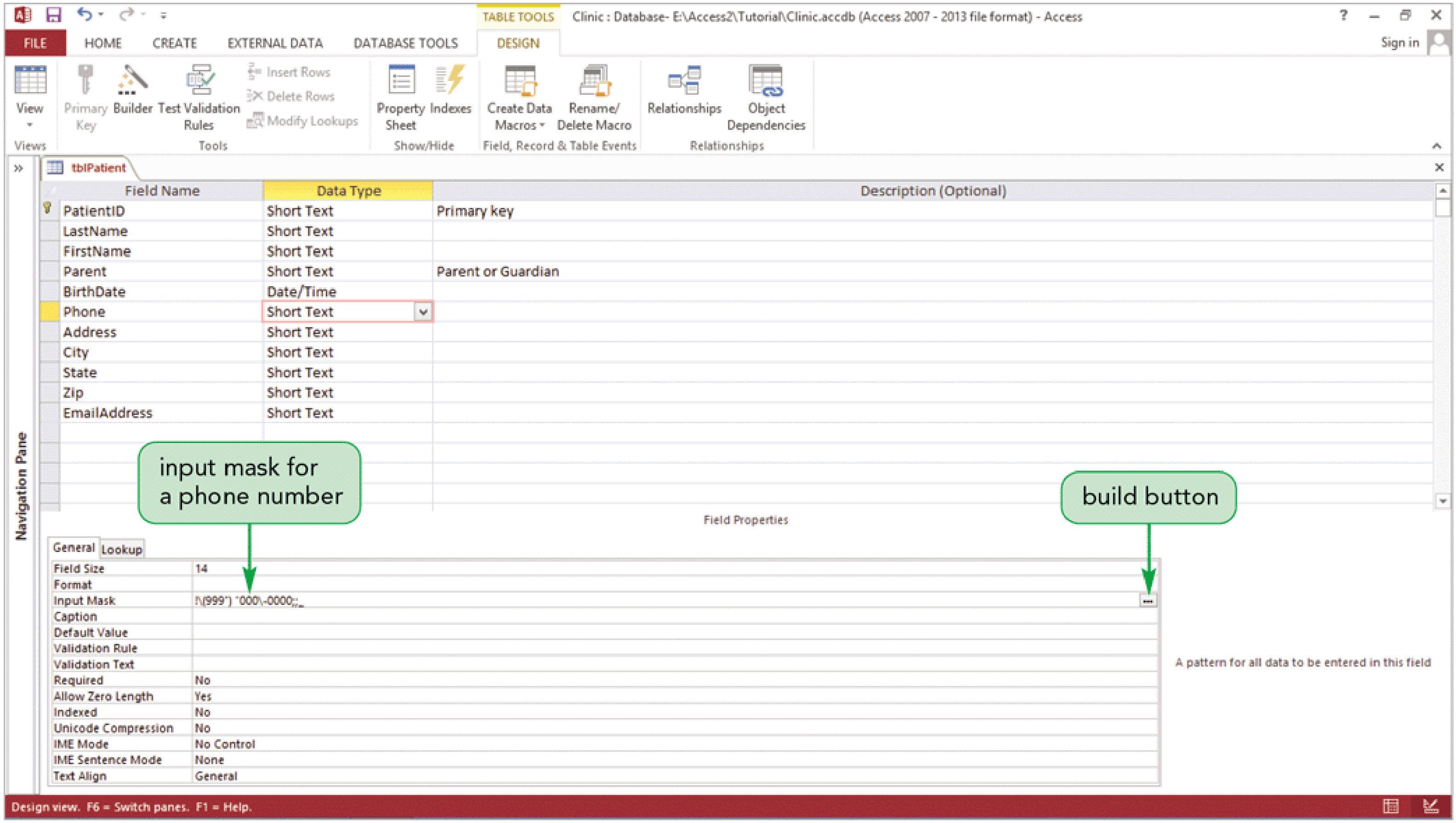
scrollable list of predefined input masks

sample values for the corresponding input masks

practice area

Using the Input Mask Wizard (Cont.)

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



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	

Field Size	14
Format	
Input Mask	!\(999") "000\-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

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

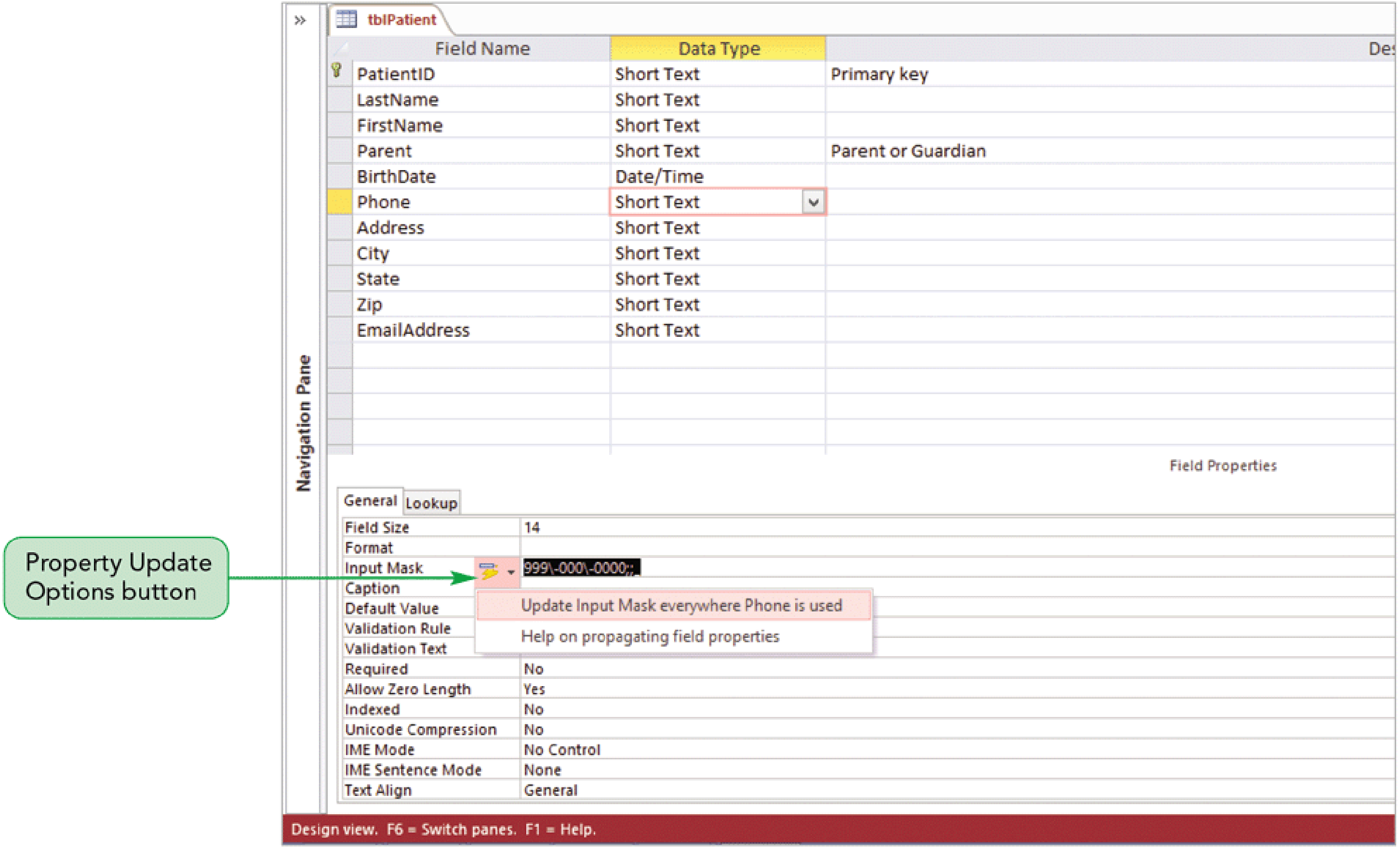
input mask for a phone number

build button

MONASH University

Using the Input Mask Wizard (Cont.)

Figure 5-36 **Property Update Options button menu**



	Field Name	Data Type	Des
🔑	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	

Field Properties

General Lookup

Field Size	14
Format	
Input Mask	999\-000\-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

Update Input Mask everywhere Phone is used
Help on propagating field properties

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

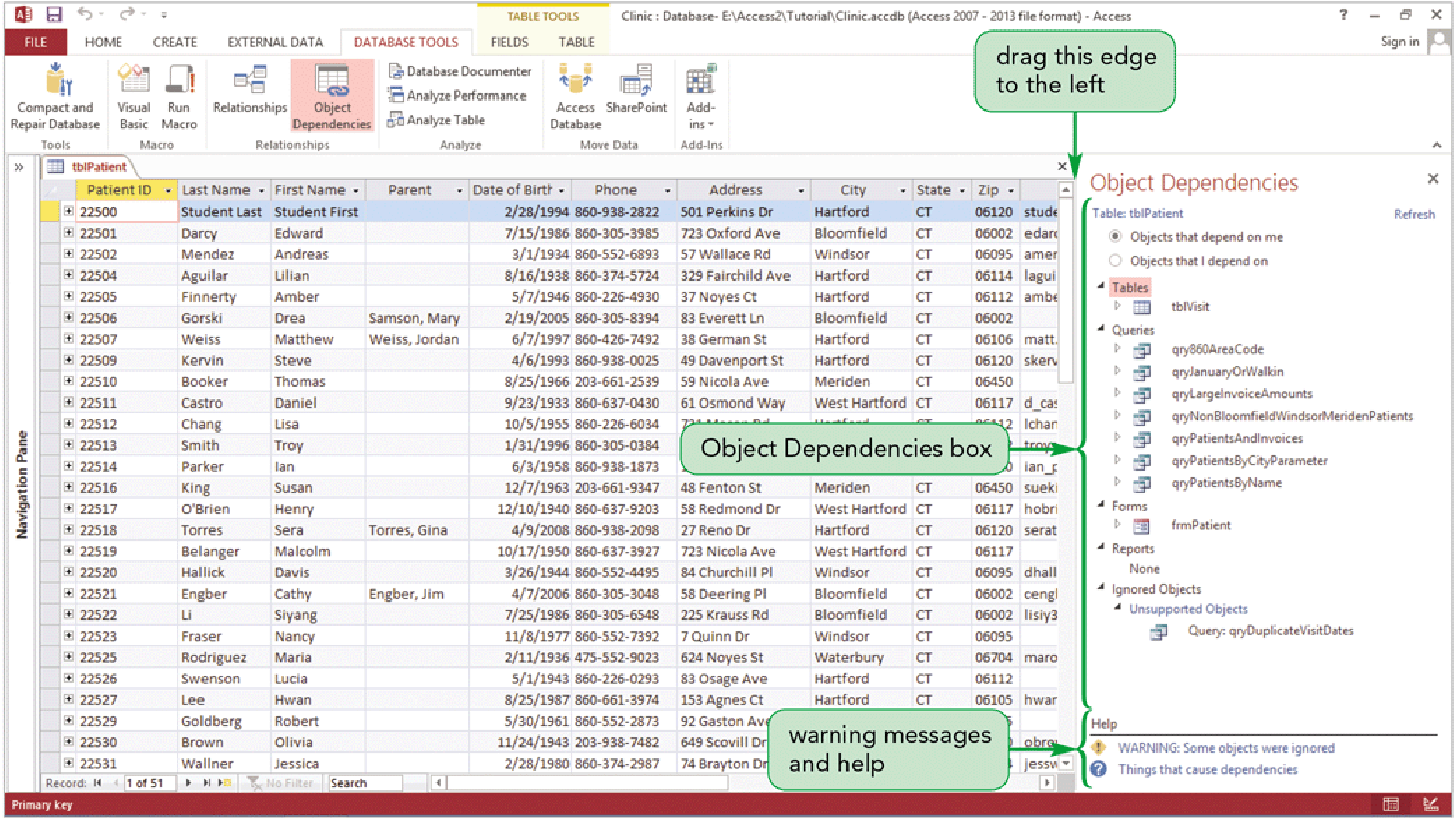
Property Update Options button

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



Patient ID	Last Name	First Name	Parent	Date of Birth	Phone	Address	City	State	Zip	
22500	Student Last	Student First		2/28/1994	860-938-2822	501 Perkins Dr	Hartford	CT	06120	stude
22501	Darcy	Edward		7/15/1986	860-305-3985	723 Oxford Ave	Bloomfield	CT	06002	edarc
22502	Mendez	Andreas		3/1/1934	860-552-6893	57 Wallace Rd	Windsor	CT	06095	amer
22504	Aguilar	Lilian		8/16/1938	860-374-5724	329 Fairchild Ave	Hartford	CT	06114	lagui
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	
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	721 Mason Rd	Hartford	CT	06112	lchan
22513	Smith	Troy		1/31/1996	860-305-0384	[illegible]	[illegible]	[illegible]	[illegible]	troy
22514	Parker	Ian		6/3/1958	860-938-1873	[illegible]	[illegible]	[illegible]	[illegible]	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	[illegible]	[illegible]	[illegible]	
22530	Brown	Olivia		11/24/1943	203-938-7482	649 Scovill Dr	[illegible]	[illegible]	[illegible]	obro
22531	Wallner	Jessica		2/28/1980	860-374-2987	74 Brayton Dr	[illegible]	[illegible]	[illegible]	jessw

Object Dependencies

Table: tblPatient Refresh

● 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

drag this edge to the left

Object Dependencies box

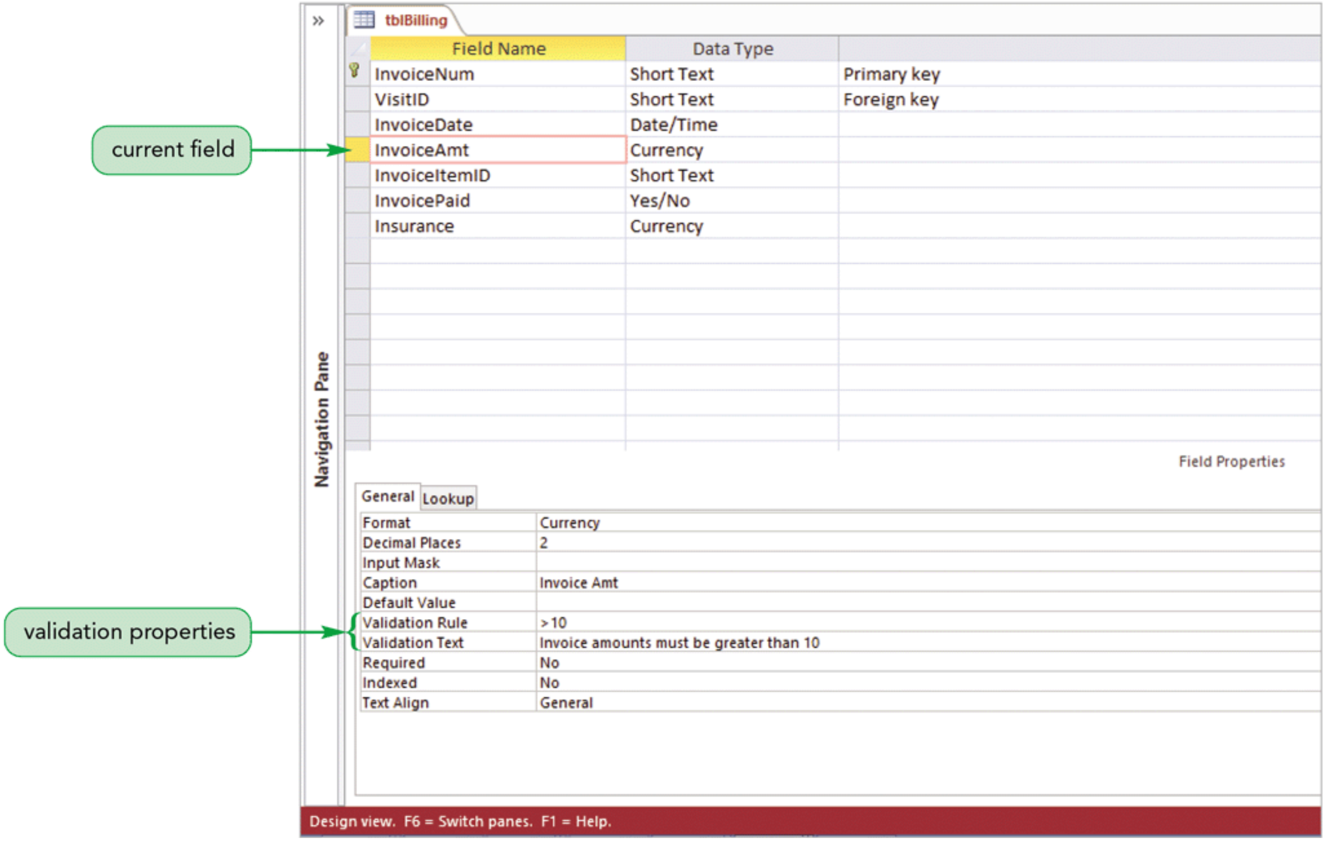
warning messages and help

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



current field

validation properties

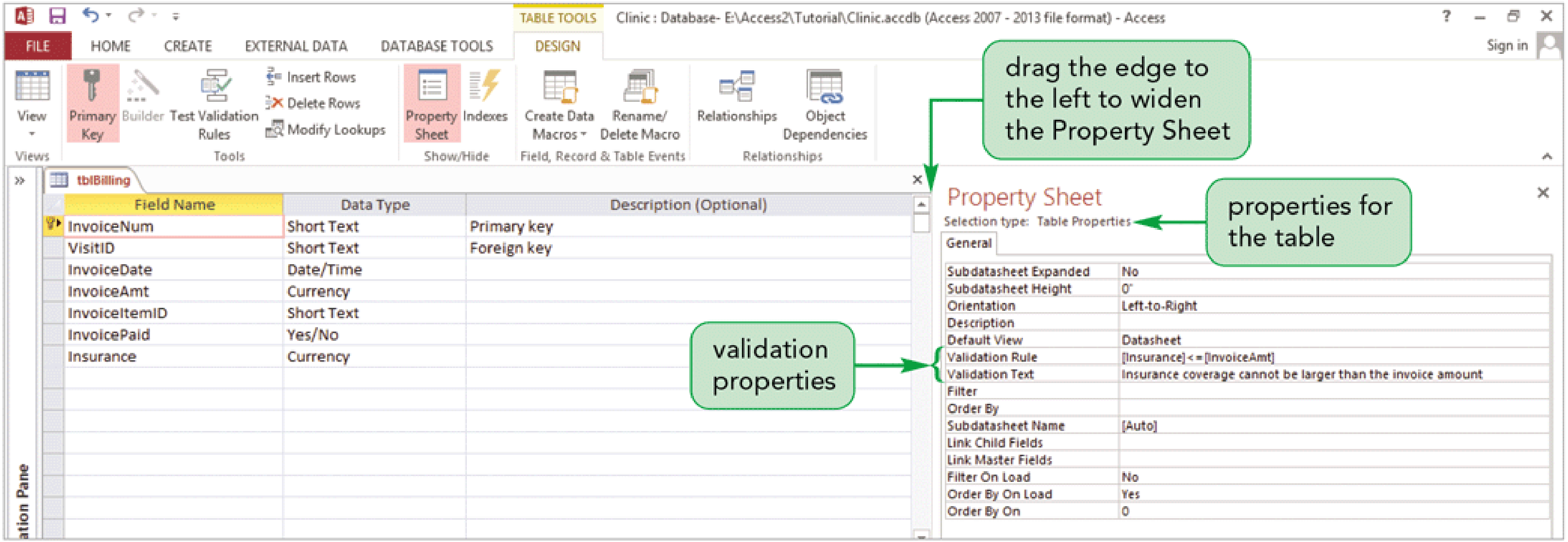
Field Name	Data Type	
InvoiceNum	Short Text	Primary key
VisitID	Short Text	Foreign key
InvoiceDate	Date/Time	
InvoiceAmt	Currency	
InvoiceItemID	Short Text	
InvoicePaid	Yes/No	
Insurance	Currency	

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



drag the edge to the left to widen the Property Sheet

properties for the table

validation properties

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

Selection type: Table Properties

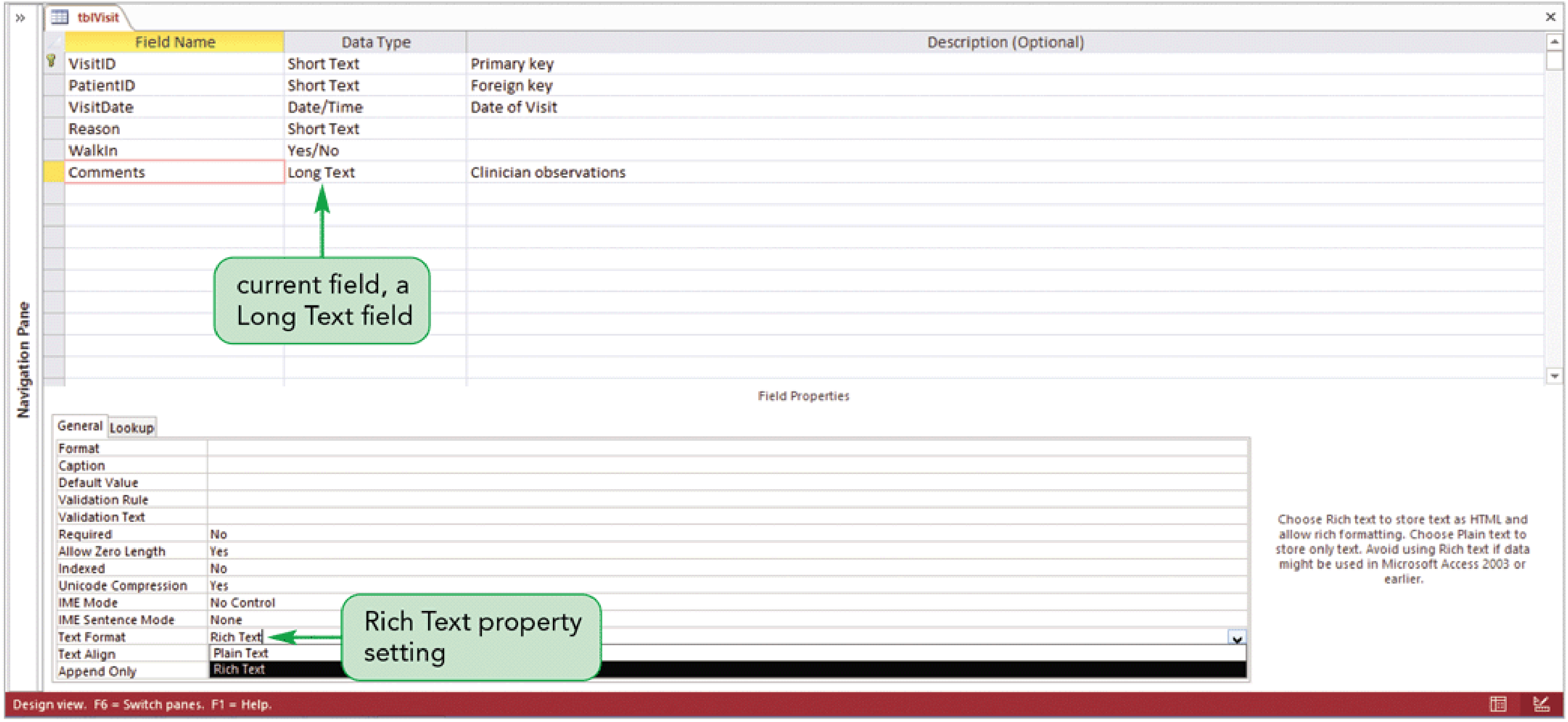
Subdatasheet Expanded	No
Subdatasheet Height	0"
Orientation	Left-to-Right
Description	
Default View	Datasheet
Validation Rule	[Insurance]<=[InvoiceAmt]
Validation Text	Insurance coverage cannot be larger than the invoice amount
Filter	
Order By	
Subdatasheet Name	[Auto]
Link Child Fields	
Link Master Fields	
Filter On Load	No
Order By On Load	Yes
Order By On	0

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



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

current field, a Long Text field

Format	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Format	Rich Text
Text Align	
Append Only	

Rich Text property setting

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.

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 Modules 3 and 5
- Read Resources in Moodle