4 Querying a Database

LESSON OUTLINE

LEARNING OBJECTIVES

After studying this lesson, you will be able to:

- Create, save, and run select queries
- Design a query using multiple tables
- Set query criteria
- Define a query sort order
- Create and format a calculated field
- Use functions in query expressions
- Create a crosstab query
- Create unmatched and duplicates queries

One goal of databases is to store data in such a way that it is easy to retrieve records. People in all types of businesses retrieve data and information daily—and often at a moment's notice. When data is stored in tables in a relational database, you can retrieve data and extract records that meet specific criteria using a *query*—a database object used to locate records based on conditions you set.

In this lesson, you will create select, crosstab, and other special queries, set query criteria, and create a form using a query. You will also create and format a calculated field, set a query sort order, set multiple query conditions, and use functions to develop a query expression. Finally, you will create special queries designed to find unmatched records between tables and find duplicate entries in a database table.

Student Resources labyrinthelab.com/acc10

CASE STUDY

Using Queries to Get Answers

After conferring with employees throughout Raritan Clinic East, James Elliott has identified several questions that nurses and other staff are frequently asked:

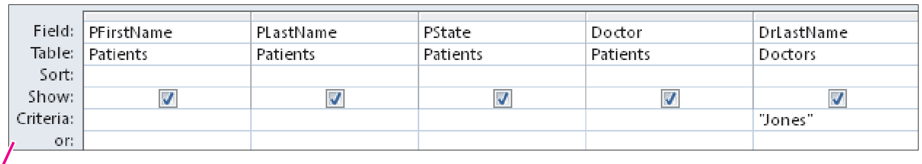
- How many current patients does a specific doctor have in the clinic?
- Who is a specific patient's doctor?
- Does anyone know a specific nurse's phone number?
- How many current patients are from states other than California?

Raritan Clinic East

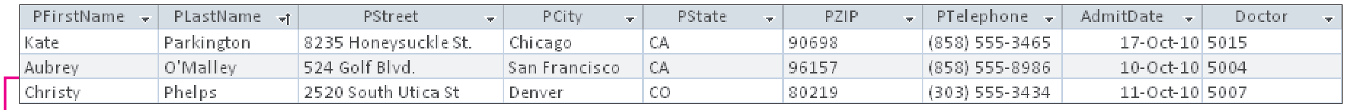
Pediatric Diagnostic Specialists

By identifying questions commonly asked by both callers and staff, James can determine the type of information that is requested of the data contained in the database. As he continues his work at Raritan Clinic East, he begins setting up queries to help human resources answer these and other questions using the database.

A query contains fields and criteria used to select records from one or more tables. In this example, the query displays fields from two database tables and searches for all records from the Patients table for Dr. Jones.

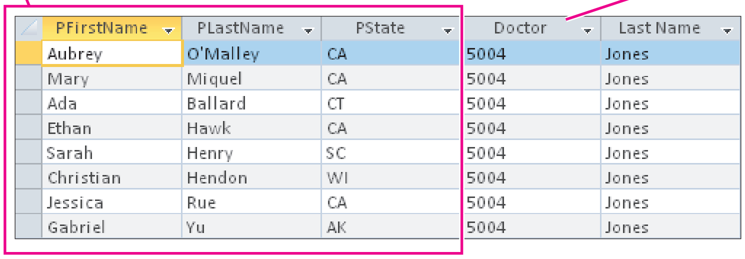


Field:	PFirstName	PLastName	PState	Doctor	DrLastName
Table:	Patients	Patients	Patients	Patients	Doctors
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					"Jones"
or:					

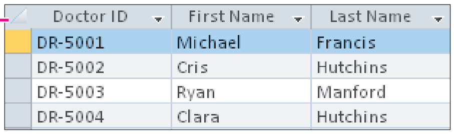


PFirstName	PLastName	PStreet	PCity	PState	PZIP	PTelephone	AdmitDate	Doctor
Kate	Parkington	8235 Honeysuckle St.	Chicago	CA	90698	(858) 555-3465	17-Oct-10	5015
Aubrey	O'Malley	524 Golf Blvd.	San Francisco	CA	96157	(858) 555-8986	10-Oct-10	5004
Christy	Phelps	2520 South Utica St	Denver	CO	80219	(303) 555-3434	11-Oct-10	5007

Patient name fields and state field come from the Patients table.



PFirstName	PLastName	PState	Doctor	Last Name
Aubrey	O'Malley	CA	5004	Jones
Mary	Miquel	CA	5004	Jones
Ada	Ballard	CT	5004	Jones
Ethan	Hawk	CA	5004	Jones
Sarah	Henry	SC	5004	Jones
Christian	Hendon	WI	5004	Jones
Jessica	Rue	CA	5004	Jones
Gabriel	Yu	AK	5004	Jones



Doctor ID	First Name	Last Name
DR-5001	Michael	Francis
DR-5002	Cris	Hutchins
DR-5003	Ryan	Manford
DR-5004	Clara	Hutchins

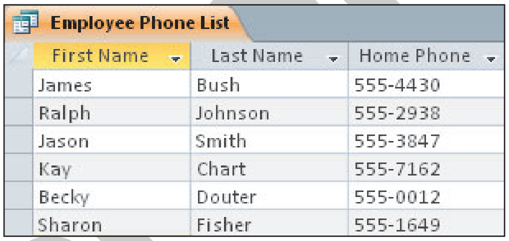
Doctor ID and last name fields come from the Doctors table. Notice that only patients whose doctor is Dr. Jones are listed in the query results datasheet.

4.1 Creating Select Queries

Video Lesson labyrinthelab.com/videos

When data is stored in tables in a relational database, you can retrieve data and extract records that meet specific criteria using a *query*—a database object used to locate records on the basis of conditions you set. Not only do queries locate records that meet specific criteria, but they also enable you to select the specific fields of data in database tables that you want to view. As a result, you can display data from selected fields in multiple tables as long as the tables are contained within the same database or linked to the database containing the query.

For example, most businesses have a telephone list of employee phone numbers so that they can reach them in time of need. By creating a select query to display only the employees' names and telephone numbers, the phone list is easy to print in a datasheet layout or report. By saving the query, you can use it again each time you need to print an updated list.


Employee Phone List

First Name	Last Name	Home Phone
James	Bush	555-4430
Ralph	Johnson	555-2938
Jason	Smith	555-3847
Kay	Chart	555-7162
Becky	Douter	555-0012
Sharon	Fisher	555-1649

A select query showing select fields from the Employees table.

Reviewing Query Features

Some important points about queries to keep in mind:

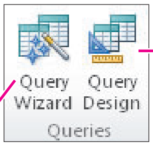
- A query acts as a saved question you ask a database or as a subset of data from one or more tables.
- Data displayed in the query results datasheet remains stored in its original table rather than in the query.
- When you edit data in a query results datasheet, you are actually editing the data stored in a table.
- Queries are dynamic objects that display up-to-date data stored in database tables.
- Queries can be used to create forms and reports containing fields from multiple tables.
- Query results datasheets enable you to filter or organize data using the same techniques you use to filter and organize table datasheets.

Identifying Tools for Creating Select Queries

The most common type of query is the *select query*. The select query retrieves data from one or more tables and displays the results in a query results datasheet. You can update records that appear in the query results datasheet and use a select query to group records and calculate sums, counts, averages, and other types of totals.

Because of the many different types of queries you may want to create, Access provides two distinct tools for creating queries:

- Query Wizard
- Design view



Query Wizard walks you through the query creation process.

Query Design enables you to create a query from scratch.

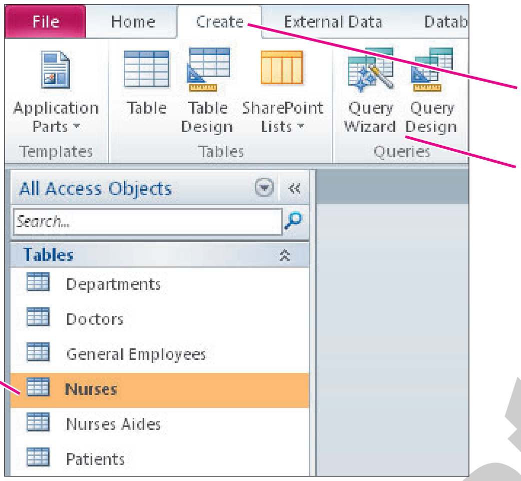
Buttons for creating queries are grouped on the Queries section of the Ribbon Create tab. You will use both tools to create queries.

DEVELOP YOUR SKILLS 4.1.1

Create a Select Query Using the Query Wizard

In this exercise, you will create a select query using the Query Wizard to create a telephone list of Raritan Clinic East nurses.

1. **Open** the Raritan Clinic East database from the Lesson 04 folder and **save** it as a new database named `Raritan Clinic East Queries`.
2. Follow these steps to activate the Query Wizard:

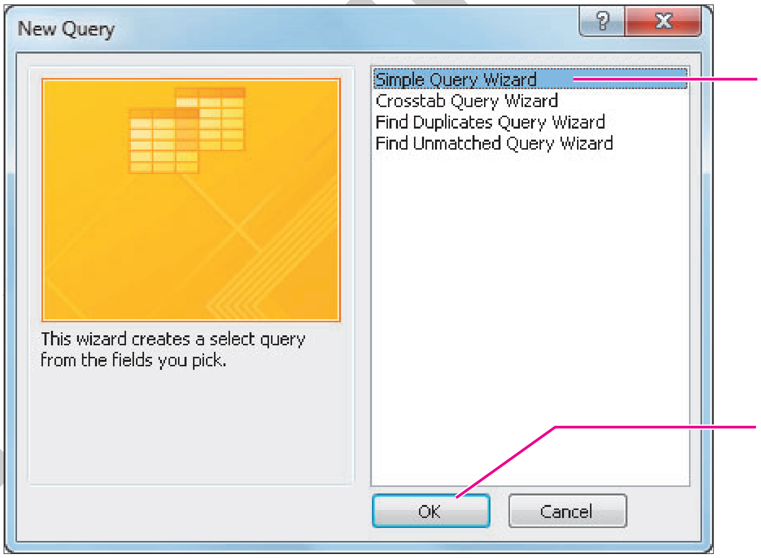


A Select the **Nurses table** in the Navigation Pane.

B Click the **Create** tab.

C Click the **Query Wizard** button in the Queries section of the Ribbon.

3. Follow these steps to select the query type:

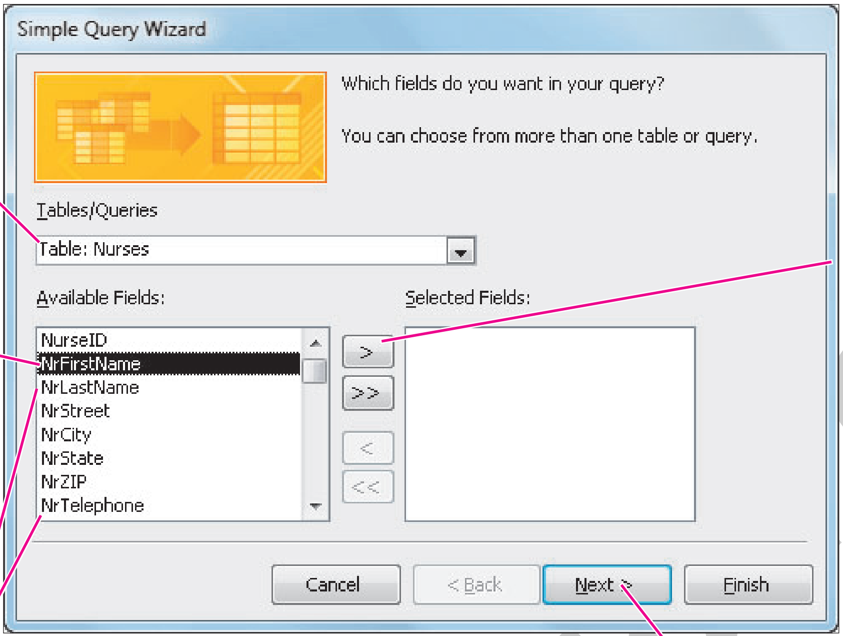


This wizard creates a select query from the fields you pick.

A Select **Simple Query Wizard.**

B Click **OK.**

4. Follow these steps to move selected fields to the query:



Which fields do you want in your query?

You can choose from more than one table or query.

Tables/Queries: Table: Nurses

Available Fields:

- NurseID
- NrFirstName
- NrLastName
- NrStreet
- NrCity
- NrState
- NrZIP
- NrTelephone

Selected Fields:

Ⓐ Ensure that the **table name** appears in the Tables/Queries box.

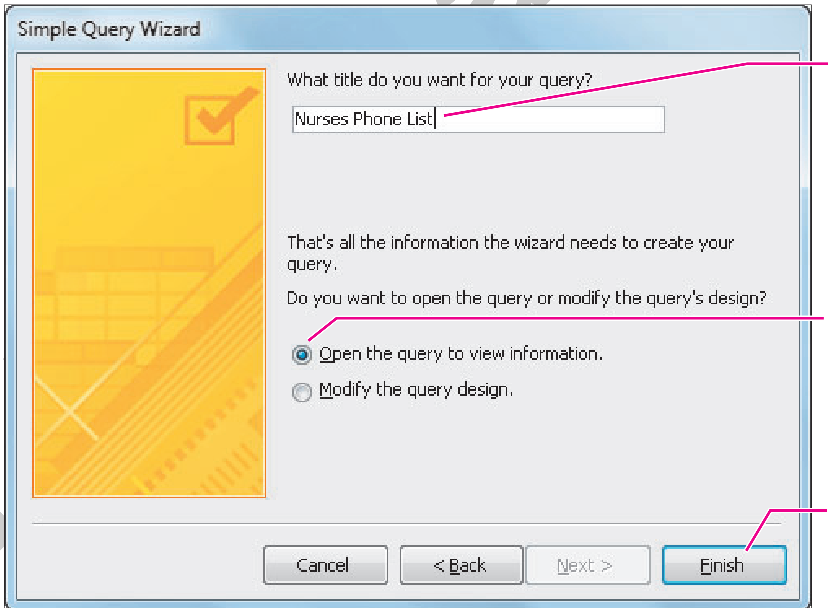
Ⓑ Select the **NrFirstName** field in the Available Fields list.

Ⓒ Click the **Move** button to move the field to the Selected Fields list.

Ⓓ **Repeat the procedures to move NrLastName and NrTelephone to the Selected Fields list.**

Ⓔ Click **Next.**

5. Follow these steps to complete the query:



What title do you want for your query?

Nurses Phone List

That's all the information the wizard needs to create your query.

Do you want to open the query or modify the query's design?

● Open the query to view information.

○ Modify the query design.

Ⓐ Type **`Nurses Phone List`** in the text box.

Ⓑ Select the **Open the Query to View Information** option.

Ⓒ Click **Finish.**

6. **Review** the query results datasheet and then **close** ☒ the query.

Creating a Select Query Using Query Design

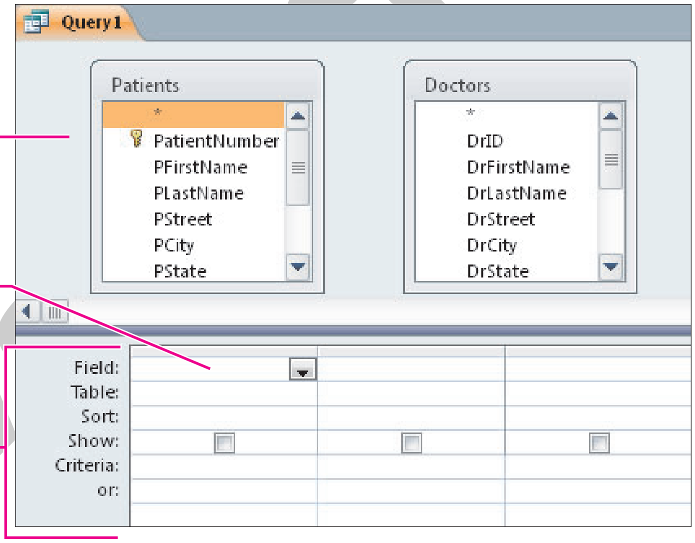
Video Lesson labyrinthelab.com/videos

The query you created in Develop Your Skills 4.1.1 simply reported the data contained in the selected fields without regard to limiting the records Access listed. When databases grow to contain voluminous records, it is often necessary to select specific records from a database by setting criteria. Using Query Design view, you will be able to take advantage of features that allow you to:

- Select fields from multiple tables
- Set criteria to locate records based on data contained in one or more fields
- Calculate totals
- Show fields containing criteria that are hidden in the query results datasheet

Identifying Features of the Query Design Grid

When you use Query Design, a grid displays that enables you to add multiple tables to the query and place fields from each table into the display grid. You can place the fields in the order in which you want them to appear in the query results datasheet and change the arrangement as necessary. In addition, the query design grid contains elements that enable you to set criteria, sort data in the query results, and so forth.



Tables containing fields to be included in the query appear in the upper pane of the Query Design window.

Fields to include in the query appear in the first row of the grid in the bottom pane.

Additional elements enable you to set Criteria, Sort data, and so forth.

The Query Design Grid

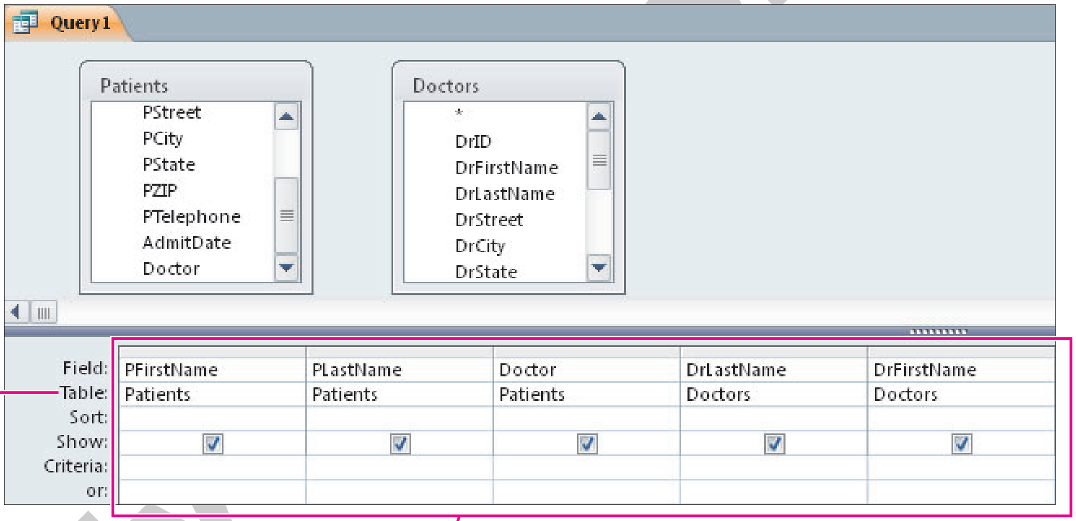
Adding Fields to the Query Design Grid

Access offers a variety of different techniques for adding fields to the query grid. These include:

- Double-clicking a field name to add the field to the next available column of the query design grid
- Dragging a field to the next column in the grid
- Clicking the Field row of a column in the query grid and selecting the field from the drop-down list
- Double-clicking the asterisk (*) that appears at the top of the field list to add all fields to the grid

When you use the asterisk to add all fields to the grid, Access places the table name in the Field row, but when you run the query, each field appears in a separate column of the query results datasheet.

- Double-clicking the field list title bar to place each field in the table in a separate column of the query grid.



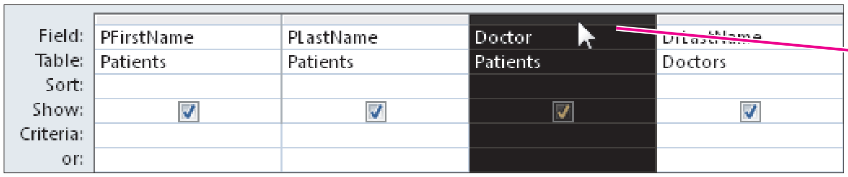
Field:	PFirstName	PLastName	Doctor	DrLastName	DrFirstName
Table:	Patients	Patients	Patients	Doctors	Doctors
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

The table containing data for the field appears in the Table row of the grid.

Each field appears in a separate column.

Rearranging Fields in the Query Design Grid

Regardless of whether the datasheet displays table data or query results data, you can rearrange the columns by dragging them and dropping them in a new position. You can also drag a column in a query grid to reposition it, when necessary. Click the narrow gray bar button that appears above a column in the query grid, and then point to the top edge of the selected column and drag it to a new position.



Field:	PFirstName	PLastName	Doctor	DrLastName
Table:	Patients	Patients	Patients	Doctors
Sort:				
Show:	☑	☑	☑	☑
Criteria:				
or:				

The Column Heading button is used to select the Field column in the query grid.

QUICK REFERENCE USING QUERY DESIGN

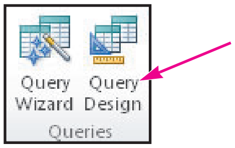
Task	Procedure
Create a query using query design	■ Choose Create→Queries→Query Design on the Ribbon.
Display query design from the query Datasheet View	■ Choose Home→Views→View on the Ribbon. *or* ■ Click the Views menu ▼ and select Design View. *or* ■ Right-click the query tab and select Design View from the shortcut menu.
Add fields to a query grid	■ Double-click a field name in the table field list. *or* ■ Drag a field from the table field list to a column of the query grid. *or* ■ Double-click the asterisk in the table field list to add all fields from the table to the query grid. *or* ■ Click Field row in the query grid and select the field from the drop-down list.
Add criteria to a query grid	■ Type criteria into the Criteria row for the field that should contain the value.
Save a query	■ Create the query and click Save. ■ Type a name for the query and click OK.
Run a query	■ Double-click a query name in the Navigation Pane. *or* ■ Create the query and display the query in Design View. ■ Choose Design→Results→Run on the Ribbon.

DEVELOP YOUR SKILLS 4.1.2

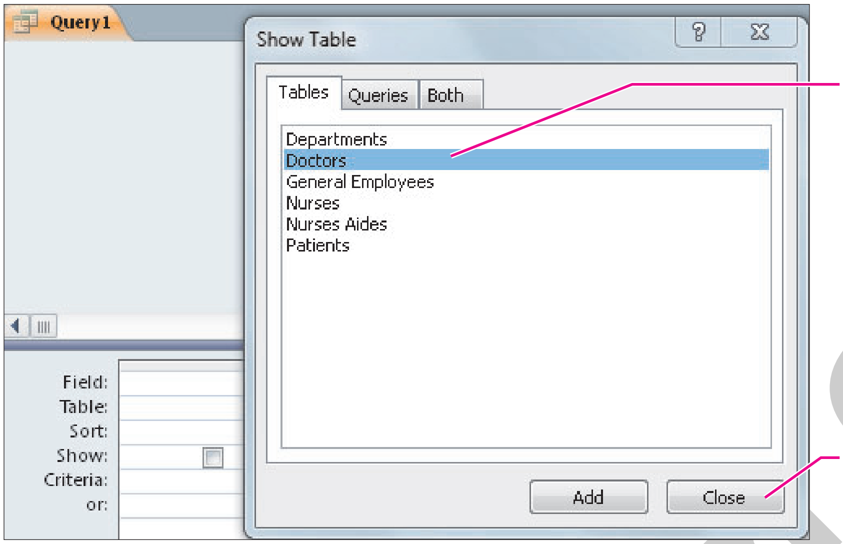
Create a Query Using Query Design

You have already created a telephone list for the Nurses and now need one for the Doctors. In this exercise, you will create a query containing fields from the Doctors table in the Raritan Clinic East Queries database, rearrange the columns in the query grid, and sort the query results list.

1. **Open** your Raritan Clinic East Queries database, if it is closed, and then choose **Create→Queries→Query Design** on the Ribbon to display the Query Design Grid.
   Access creates a new query, displays the query grid, and opens a list of tables contained in the database so that you can choose the tables you want to include in the query.



2. Follow these steps to add the Doctors table to the query:



A **Double-click** the Doctors table.

B **Click Close.**

Notice that the table name appears in the field list title bar.

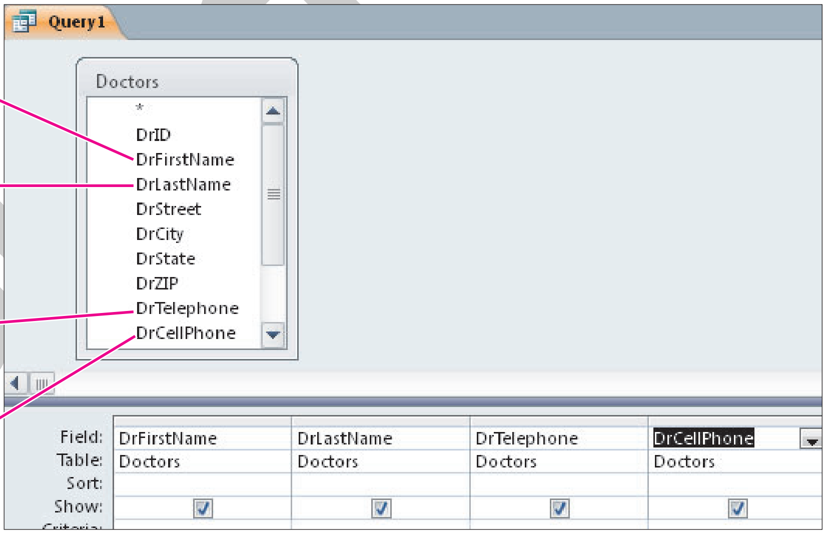
3. Follow these steps to add fields to the query grid:

A **Double-click** DrFirstName to add it to the query grid first column.

B **Double-click** DrLastName to add it to the query grid second column.

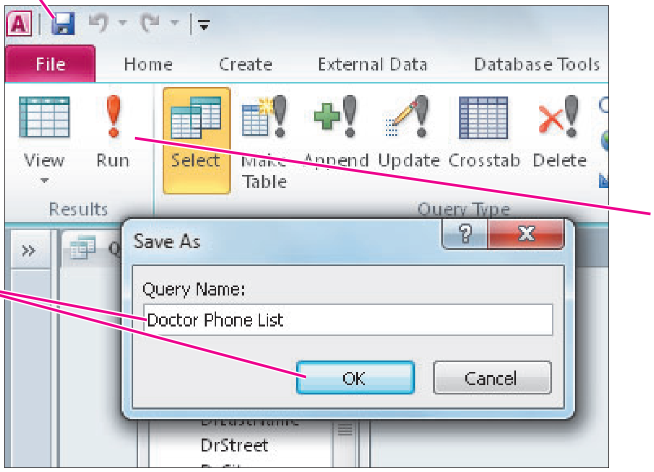
C **Double-click** the DrTelephone to add it to the query grid third column.

D **Double-click** the DrCellPhone to add it to the query grid fourth column.



Field:	DrFirstName	DrLastName	DrTelephone	DrCellPhone
Table:	Doctors	Doctors	Doctors	Doctors
Sort:				
Show:	☑	☑	☑	☑

4. Follow these steps to save and then run the query:



Query Name: Doctor Phone List

A Click **Save.**

B Type **Doctor Phone List** in the Query Name text box and click **OK.**

C Choose **Design→Results→Run** on the Ribbon.

Access runs the query and displays two columns of data for all records in the database tables.

5. **Press** [Tab] to advance to the Last Name field column and then choose **Home→Sort & Filter→Ascending** to sort the list alphabetically by last name.

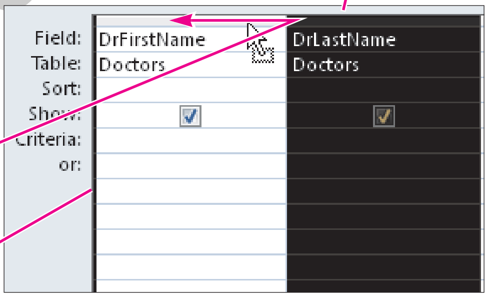
6. Choose **Home→Views→View** on the ribbon to switch to Design view.

7. Follow these steps to move the DrLastName field to the left of DrFirstName:

A **Click** the narrow grey border above the DrLastName column in the query grid. The mouse pointer should appear as a solid black down-pointing arrow when you point to the correct grey border position.

B **Drag** the selected column grey border to the left of the DrFirstName column. To drag the selected column, the mouse pointer should appear as a hollow white arrow.

C **Drop** the column when the dark black vertical line appears as shown here.



Field:	DrFirstName	DrLastName
Table:	Doctors	Doctors
Sort:		
Show:	☑	☑
Criteria:		
or:		

8. Choose **Design→Results→Run** to run the query again.

9. **Close** the query, **saving** changes when prompted.

Designing a Query Using Multiple Tables

Video Lesson labyrinthelab.com/videos

Until now, all the datasheets you have seen have displayed data from only one table. There will be times when you want to view data that is contained in different tables within or linked to the same database. Queries make displaying data from multiple tables possible.

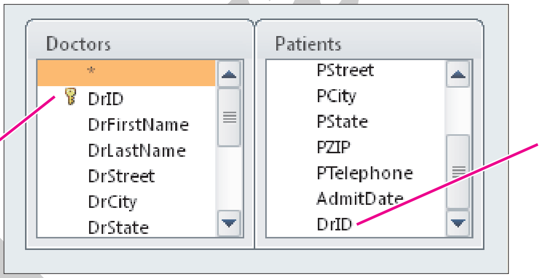
Choosing Fields to Include in a Query

When you build a query, you select only those fields that you want to display in the query results datasheet—leaving out those fields that have no impact on the data you want to view or that are confidential. For example, if you were responsible for maintaining a list of FBI agents, would you want everyone with access to the database to know the addresses and phone numbers of all agents? By selecting only specific fields of data in database tables and displaying those fields in a query, you can use the query as the basis for creating a report or form, thus protecting the confidential data. When the data you want to display in a query results datasheet appears in different tables in the database, you simply add those tables to the query design grid to make fields from the tables available.

Selecting a Field Appearing in Multiple Tables

Many times, as you work with table field lists, you will use tables containing the same field names and wonder which field you should add to a query. There are various ways to determine which field list to use for a field:

- Review the table data for the tables containing the same field name. In many cases, you will be able to determine which table field to use based on the query results you want.
- Identify the table for which a field is the primary key. In many cases, the primary key will enable you to pull other data contained in the table. For example, if you want to use the Customer Number field to access a customer name, you would use the Customer Number field from the table that also contains the customer name.



DrID is the primary key field in the Doctors table.

DrID appears in the Patients table but is not a primary key field.

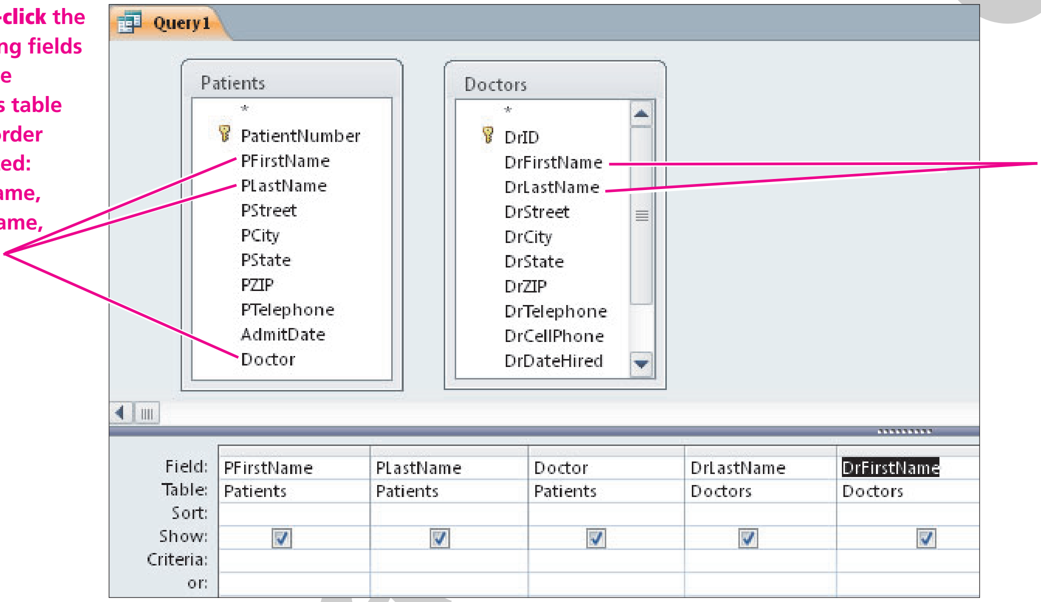
DEVELOP YOUR SKILLS 4.1.3

Create a Multi-Table Query

In this exercise, you will create a multi-table query using Query Design View that displays patient names from the Patients table and their doctors' names from the Doctors table.

Before You Begin: Your Raritan Clinic East Queries database should be open.

1. Choose **Create→Queries→Query Design** on the Ribbon.
2. **Double-click** the following table names in the Show Table dialog box to add the table field lists to the upper pane of the query: Patients and Doctors.
3. **Close** the Show Table dialog box, and then follow these steps to add fields to the query grid:



Field:	PFirstName	PLastName	Doctor	DrLastName	DrFirstName
Table:	Patients	Patients	Patients	Doctors	Doctors
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

A **Double-click** the following fields from the Patients table in the order presented: PLastName, PFirstName, Doctor.

B **Double-click** the following fields from the Doctors table in the order presented: DrLastName, DrFirstName.

The fields should appear in the query grid in the order shown in the figure.

4. **Save** the query using the query name **`Patients & Doctors`**.
5. Choose **Design→Results→Run** on the Ribbon to display the query results.
6. **Close** the query results datasheet.

4.2 Setting Query Criteria

Video Lesson labyrinthelab.com/videos

As you begin working with large databases that contain hundreds of thousands of records, you experience the power behind queries that enables you to specify *criteria*—conditions that data must meet. When you run the query, Access will list only those records containing data that meet the criteria. This is the feature used by sportscasters, telephone order clerks, and others who need data instantly.

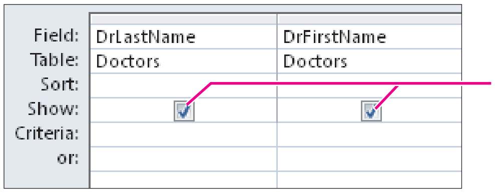
Adding Criteria to a Query

The conditions and comparison format used in queries to limit or control the number of records Access finds resemble the criteria you set when you filter tables by form. The same comparison operators (<, >, =, >=, <=, <>) used in setting validation rules help define criteria in queries.Setting criteria limits the number of records displayed in a query results datasheet to only those records with values in the selected field columns that meet the criteria. In addition, the following comparison and logical criteria can be used to limit data returned in queries:

Criteria Expression	Description
> 123	For a numeric data field, returns records for values greater than 123.
< 100.45	For a currency data field, values less than 100.45.
>= Smith	For a text data field, all values from Smith through the end of the alphabet.
Between 2/2/1999 And 12/1/1999	For date data field, dates from February 2, 1999 through December 1, 1999.
Not Smith	For a text data field, all records for values except Smith.
Not 2	For numeric data field, all values except those equal to 2.
Not T*	For text data field, all values that don't start with the letter T.
In("Canada", "UK")	For a text data field, records containing the values Canada or UK in the criteria field.
"London" Or "Hedge End"	For a text data field, orders shipped to London or Hedge End.
Date()	For a date data field, values for today's date.
Between Date( ) And DateAdd ("M", 3, Date( ))	For a date data field, values required between today's date and three months from today's date.
< Date( ) – 30	For a date data field, values 30 days prior to the current date.

Hiding Columns in the Query Results Datasheet

Sometimes you will need to enter criteria in a field contained in a query grid and display values from a different field. The Show checkbox in the query grid enables you to hide or show fields in the query results datasheet by simply checking or clearing the check in the Show box.



Field:	DrLastName	DrFirstName
Table:	Doctors	Doctors
Sort:		
Show:	☑	☑
Criteria:		
or:		

The Show box appears for each field in the query grid so that you can check those you want to display in the query results datasheet.

Saving Queries with Criteria

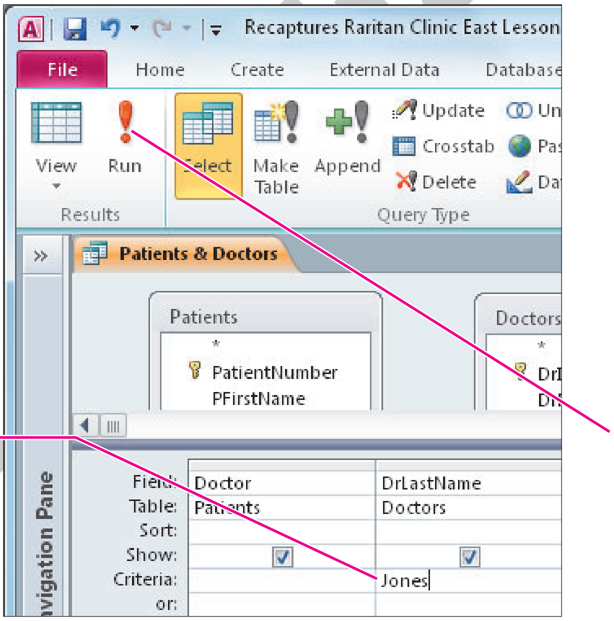
When you add criteria to a query to locate specific records in the database, Access recognizes the criteria as a change in the query design. As a result, when you close the query, Access prompts you to save it. Saving the query saves the criteria as part of the query. Sometimes, when you find yourself setting the same criteria for a query time and time again, you may want to save the query with the criteria. Many times, however, you will want to set different criteria each time you run the query. In those cases, you would want to respond to the save prompt by choosing No.

DEVELOP YOUR SKILLS 4.2.1

Add Criteria and Run a Query

In this exercise, you will add criteria to the query grid and run the query.

1. **Open** the Navigation Pane and **double-click** the Patients & Doctors query to run the query.
2. Choose **Home→Views→View** on the Ribbon to display Query Design View.
3. Follow these steps to add criteria to the query grid:



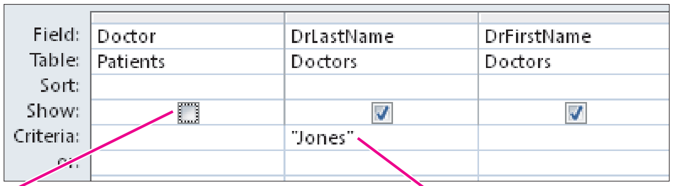
Field:	Doctor	DrLastName
Table:	Patients	Doctors
Sort:		
Show:	☑	☑
Criteria:		Jones
or:		

A Click in the **Criteria** row of the query grid for the DrLastName field and type Jones.

B Choose **Design→Results→Run** on the Ribbon to run the query.

Access runs the query and searches the database for all Patients for Dr. Jones.

4. Choose **Design→Views→View** to return to Design View and then follow these steps to remove a column from display:



Field:	Doctor	DrLastName	DrFirstName
Table:	Patients	Doctors	Doctors
Sort:			
Show:	□	☑	☑
Criteria:		"Jones"	
or:			

Ⓐ Click the **Show** box for Doctor to clear the checkmark.

Ⓑ Notice that Access placed quotation marks around the text you entered in the Criteria row to show that it is text that should appear as shown.

5. Choose **Design→Results→Run** and review the results datasheet.

6. **Close** the query. Choose **No** when prompted to save changes to the query.
Saving changes to the query now saves the new field as part of the query. It also saves the criteria as part of the query. Because you plan to use the query to set criteria to display records for other patients, you will need to remove the existing criteria before running it for another patient.

Using Wildcards

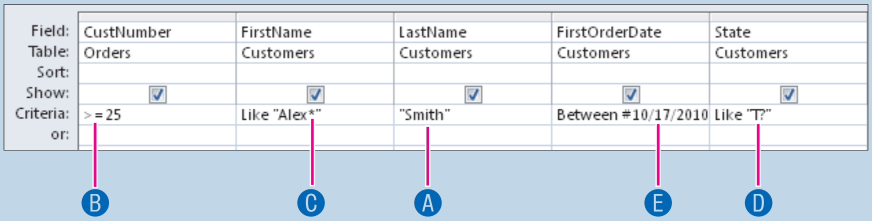
Video Lesson labyrinthelab.com/videos

Wildcards—symbols designed to represent any character or a group of characters—make locating records that contain specific text or values more efficient, especially when you are uncertain of the spelling of a name or want to select records containing variable data in a field. Access uses two primary wildcards, as described in the following Quick Reference table.

QUICK REFERENCE DEFINING WILDCARD SYMBOLS

Symbol	Description of Use
An asterisk (*)	Substitutes for a group of characters that appears at the position of the asterisk **Example**: If you type **`R*`** in the last name column of a query grid, Access will locate all names beginning with *R* regardless of how many characters make up the name—*Rogers, Rich,* and *Rovarino* would all appear in the results datasheet.
A question mark (?)	Substitutes for a single character that might appear at the position of the question mark **Example**: If you type **`m?s`** in the criteria row for a column, Access will locate records containing values such as *mrs, ms, mbs*, and so forth.
Open/close brackets [ ]	Matches text or individual characters placed within the brackets individually. **Example:** If you type ca[rt], Access will find cat and car but not cab or cad, etc.
Exclamation point (!)	Matches any character within the brackets *except* those characters that follow the !. **Example:** If you type ca[!rt], Access will find cab, cad, cam, etc., but *not* cat or car.
Hyphen (-)	Matches characters at the wildcard position that fall within a range of ascending values. **Example:** If you type ca[a-r], Access finds cab, cad, cam, car, etc., but *not* cat or cay.
Number sign (#)	Locates any numeric digit at the position of the #. **Example:** If you type #10, Access locates 010, 110, 210, etc.

Examples of query criteria are shown in the following figure.



Field:	CustNumber	FirstName	LastName	FirstOrderDate	State
Table:	Orders	Customers	Customers	Customers	Customers
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:	>=25	Like "Alex*"	"Smith"	Between #10/17/2010	Like "T?"
or:					

Item	Query Criterion	What It Does
A	Value	Tells Access to display records with this specific value.
B	Comparison Operators	Tells Access to locate a range of values for this field.
C	Wildcard	Tells Access to locate items beginning with specific characters regardless of how many characters follow.
D	Wildcard	Tells Access to locate items beginning with this character with one character following the T.
E	Dates	Tells Access to display records with dates within a specific date range.

Setting AND and OR Criteria

In some cases, as you query a database, you need to select records that meet multiple criteria. Access uses two basic criteria conditions that apply to setting multiple criteria—AND and OR. The basic principles for determining whether to use AND or OR criteria in queries are as follows:

QUICK REFERENCE **USING *AND* AND *OR* CRITERIA**

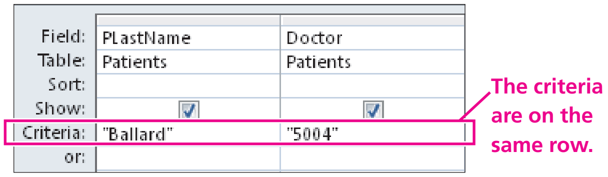
Criterion Type	Description
AND	Use to select records that meet all criteria set in all query grid fields. For example, you might set AND criteria to locate customers from a specific city within a state by typing the city name in the City field on the query grid and the state in the State field on the grid.
OR	Use to select records meeting one condition or another condition whether the criteria are set for the same field or different fields. For example, you could set OR criteria to locate customers from two different states. By setting OR criteria, Access displays all records containing one state *or* the other state in the State field.

Positioning Multiple Criteria in the Query Grid

In the query grid, AND criteria all appear on the *Criteria* row even when criteria are set for different fields. The word *And* appears between values in the same field. When you set OR criteria, the first criterion is entered on the *Criteria* row of the grid while other criteria appear on the *or* row of the grid.

Example – AND Criteria

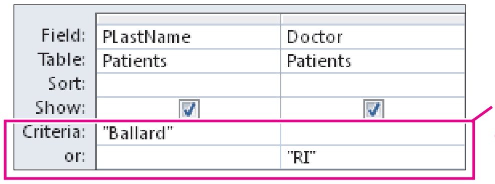
Setting criteria for two different fields on the *Criteria* row creates an AND condition. With this type of criterion, Access locates only those records for people whose last name is Ballard and whose doctor is Dr. Jones (5004). People whose last name is Ballard who have a different doctor will not appear when you run the query.



Field:	PLastName	Doctor
Table:	Patients	Patients
Sort:		
Show:	☑	☑
Criteria:	"Ballard"	"5004"
or:		

The criteria are on the same row.

Example – OR Criteria



Field:	PLastName	Doctor
Table:	Patients	Patients
Sort:		
Show:	☑	☑
Criteria:	"Ballard"	
or:		"RI"

Criteria are on different rows.

Using the same values, setting 5004 on the *or* row of the query grid tells Access to look for all patients named Ballard in the table or anyone whose doctor is Dr. Jones (5004). Patients who have other doctors will be shown if their last name is Ballard, and Dr. Jones' patients will be shown regardless of their last name.

Using AND and OR Criteria

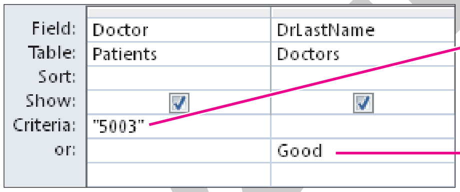
Sometimes setting AND and OR criteria seems to operate backwards. For example, if you want to locate all records for patients from TX or LA, you use an OR condition. The query results shows more listings than if you type only one value in the grid—all records for patients from both TX and LA. If you consider that there are no records that contain both TX and LA in the *State* field, it begins to make sense.

DEVELOP YOUR SKILLS 4.2.2

Use Wildcards and Multiple Criteria in Queries

In this exercise, you will set multiple criteria in queries and also use wildcards to locate variable data in records in the Raritan Clinic East Queries database.

1. **Right-click** the Patients & Doctors query in the Navigation pane and choose **Design View**.
2. Follow these steps to set Or criterion using two values for two fields:



Field:	Doctor	DrLastName
Table:	Patients	Doctors
Sort:		
Show:	☑	☑
Criteria:	"5003"	
or:		Good

A Click the **Criteria** row for the Doctor column and type **5003**.

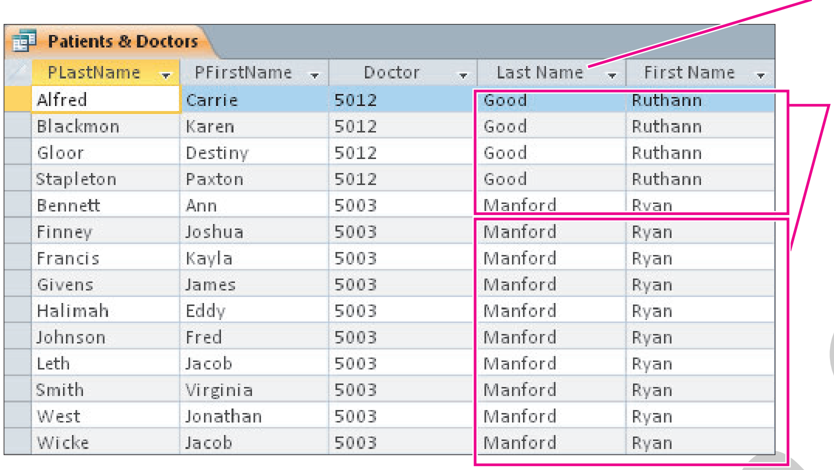
B Click the **Or** row for the DrLastName column and type **Good**.

These criteria will locate all records for doctor #5003 as well as all records for Dr. Good.

3. Choose **Design→Results→Run** on the Ribbon to run the query.

4. Review the query results datasheet:

Ⓐ Access displays a total of 14 records arranged alphabetically by Last Name.



Patients & Doctors

PLastName	PFirstName	Doctor	Last Name	First Name
Alfred	Carrie	5012	Good	Ruthann
Blackmon	Karen	5012	Good	Ruthann
Gloor	Destiny	5012	Good	Ruthann
Stapleton	Paxton	5012	Good	Ruthann
Bennett	Ann	5003	Manford	Ryan
Finney	Joshua	5003	Manford	Ryan
Francis	Kayla	5003	Manford	Ryan
Givens	James	5003	Manford	Ryan
Halimah	Eddy	5003	Manford	Ryan
Johnson	Fred	5003	Manford	Ryan
Leth	Jacob	5003	Manford	Ryan
Smith	Virginia	5003	Manford	Ryan
West	Jonathan	5003	Manford	Ryan
Wicke	Jacob	5003	Manford	Ryan

Ⓑ Access locates four records for Dr. Good and 10 records for doctor #5003 (Dr. Manford).

5. Choose **Home→Views→View** on the Ribbon to return to Query Design View.

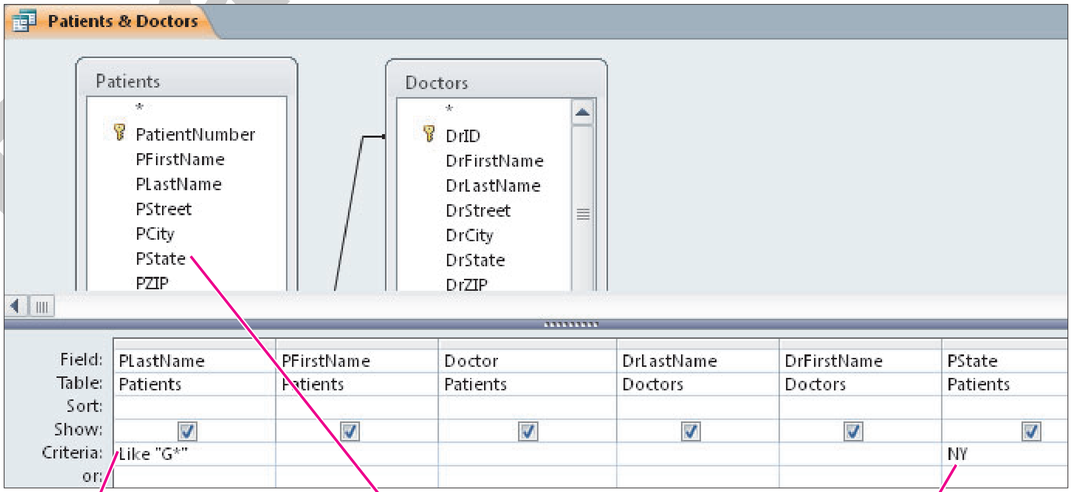
6. **Clear** the criteria in the Or row for DrLastName and then type **`Francis`** in the Or row for PLastName.

7. Choose **Design→Results→Run** on the Ribbon to run the query.
Access locates 11 records—one for a patient with the last name of Francis and ten for Dr. Manford. Do you see the record for Dr. Manford's patient with the last name of Francis?

8. Choose **Home→Views→View** on the Ribbon to return to Query Design View.

Use Wildcards

9. **Clear** all criteria from the Criteria and Or rows and then follow these steps to set different criteria:



Field:	PLastName	PFirstName	Doctor	DrLastName	DrFirstName	PState
Table:	Patients	Patients	Patients	Doctors	Doctors	Patients
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:	Like "G*"					NY
or:						

Ⓐ Type **G*** in the Criteria row for the PLastName field and then press Tab. When you move to a different column, Access adds *Like* and quotation marks around the expression.

Ⓑ **Double-click** the PState field in the Patients table to add it to the query.

Ⓒ Type **NY** in the Criteria row of the PState column.

This criteria locates all patients from NY whose last name begins with G, regardless of what other characters might be part of the last name.

10. Choose **Design→Results→Run** on the Ribbon to run the query.
    Access locates 1 record—for the NY patient whose last name begins with G.

11. **Close** ☒ the query without saving changes. Choose **No** when prompted to save changes.
    Saving changes at this point would store the criteria with the query.

Entering Date Criteria

Video Lesson labyrinthelab.com/videos

The "plain English" approach to entering criteria in Access queries can make comparing date values more logical. You can set date criteria to determine age, expired licenses, and so forth. Access acknowledges the same comparison criteria for performing date comparisons that it does for locating other types of data—regardless of the format used to enter dates.

SAMPLES OF DATE CRITERIA

Criterion	Locates
11/22/2006	Locates records containing the specific date
<22-Nov-2006	Locates records containing dates that occur before the specific date—regardless of how the date is typed
>11/22/06	Locates records containing dates that occur after the specific date
<=#11/22/06#	Locates records containing dates on or before the specific date; the # signs that appear before and after the date help Access identify the data between them as a date
Between 11/22/06 and 11/22/08	Locates records containing dates after the first date and before the second date

DEVELOP YOUR SKILLS 4.2.3

Use Date Criteria in Queries

A list of new admissions is printed each day and given to the information desk for easy reference when answering phone calls. In this exercise, you will create a new query to set criteria for locating records using date values.

1. Choose **Create→Queries→Query Design** .
2. **Double-click** the Patients table and then the Doctors table in the Add Table dialog box to add the table field lists to the query and **close** the Show Table dialog box.
3. **Double-click** each of the following fields in the Patients table to add the fields to the query grid: PFirstName, PLastName, AdmitDate.
4. **Double-click** the *DrLastName* field in the Doctors table to add the field to the query grid.
5. Click the **Save** button, type **`Daily Admissions`** in the Query Name field, and then click **OK** to save the query.
6. Type **`January 15, 2010`** into the *Criteria* row for the AdmitDate field column.
   Regardless of whether you type the date as entered here or as 01/15/10 or 1-15-2010, Access formats the date after you enter it so that it appears as #1/15/2010#.

7. Choose **Design→Results→Run** on the Ribbon to run the query.
   Access locates two records for patients admitted on January 15, 2010.
8. **Close** the query, without saving changes when prompted.

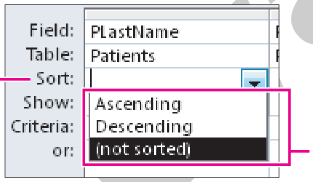
4.3 Sorting a Query and Limiting Results

Video Lesson labyrinthelab.com/videos

The sorting techniques you may already know can also be used to sort query results datasheets in ascending or descending order after you run the query. In addition, the query grid contains a *Sort* row that you can set so that Access sorts data in the query results datasheet as it runs the query. The same sort orders—ascending and descending—are available in the query grid Sort listings. Sorting data as you run the query ensures consistency in data organization to make locating data in the query results datasheet more efficient. When you work with large volumes of data, limiting the number of records shown in the query results datasheet can also be helpful. Access contains tools that enable you to limit the number of records displayed.

Setting a Query Sort Order

Because the sort order for a query varies according to the information you want to retrieve, creating a copy of a query and saving each query using a different sort order can also be beneficial. In addition, there will be times when you want to sort data on two fields so that when the data in one field (such as *LastName* or *State*) is the same, Access looks at the second field (such as *FirstName* or *City*) to sort. When two fields are set as sort fields, Access sorts the fields left to right as they appear in the query grid. The first field found with a sort order identified is the primary sort field; the next field containing a sort order is the secondary sort field. There is no need to arrange the columns in the query grid side by side to sort multiple fields.



Field: PLastName
Table: Patients
Sort:
Show:
Criteria:
or:

The Sort row of the query grid.

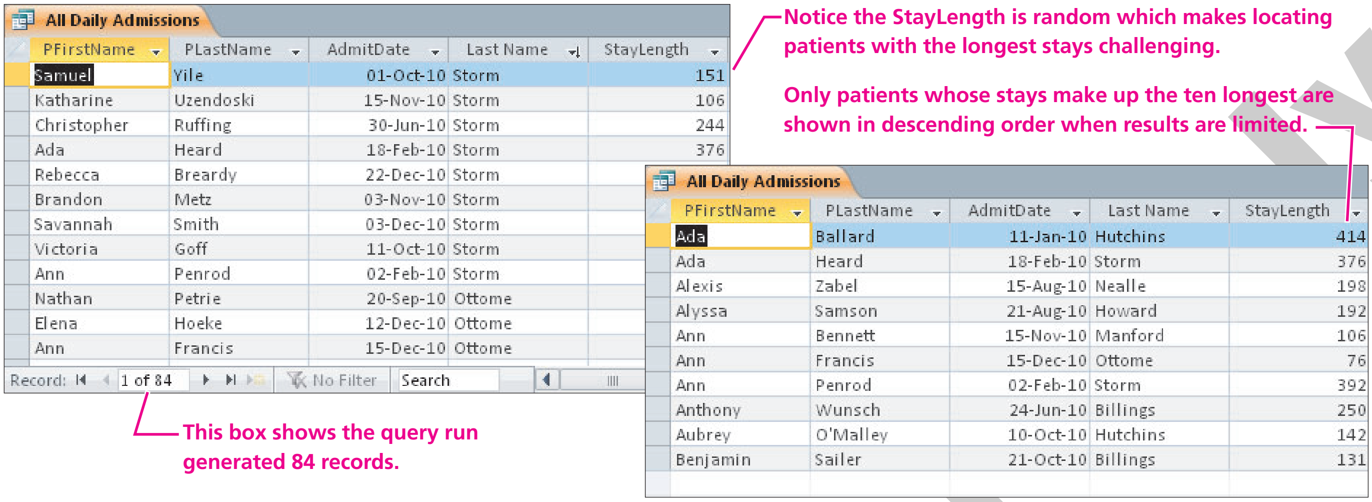
Sort orders available.

Multi-valued lookup fields cannot be used as sort fields.

Limiting Number of Results Displayed

Running queries on large databases containing many tables and hundreds of thousands of records often returns such a large number of results when you run a query that it can be challenging to filter out the data records you were looking for. Limiting the number of records Access displays when you run a query can be beneficial, especially when combined with sorting features.

For example, if you set up a query to sort the query results in descending order and then limit the number of items displayed to ten, you would, if effect, have a list of the top ten items in the table or tables being queried.



All Daily Admissions

PFirstName	PLastName	AdmitDate	Last Name	StayLength
Samuel	Yile	01-Oct-10	Storm	151
Katharine	Uzendoski	15-Nov-10	Storm	106
Christopher	Ruffing	30-Jun-10	Storm	244
Ada	Heard	18-Feb-10	Storm	376
Rebecca	Breardy	22-Dec-10	Storm	
Brandon	Metz	03-Nov-10	Storm	
Savannah	Smith	03-Dec-10	Storm	
Victoria	Goff	11-Oct-10	Storm	
Ann	Penrod	02-Feb-10	Storm	
Nathan	Petrie	20-Sep-10	Ottome	
Elena	Hoeke	12-Dec-10	Ottome	
Ann	Francis	15-Dec-10	Ottome	

Record: 1 of 84 No Filter Search

Notice the StayLength is random which makes locating patients with the longest stays challenging.

Only patients whose stays make up the ten longest are shown in descending order when results are limited.

All Daily Admissions

PFirstName	PLastName	AdmitDate	Last Name	StayLength
Ada	Ballard	11-Jan-10	Hutchins	414
Ada	Heard	18-Feb-10	Storm	376
Alexis	Zabel	15-Aug-10	Nealle	198
Alyssa	Samson	21-Aug-10	Howard	192
Ann	Bennett	15-Nov-10	Manford	106
Ann	Francis	15-Dec-10	Ottome	76
Ann	Penrod	02-Feb-10	Storm	392
Anthony	Wunsch	24-Jun-10	Billings	250
Aubrey	O'Malley	10-Oct-10	Hutchins	142
Benjamin	Sailer	21-Oct-10	Billings	131

This box shows the query run generated 84 records.

The Return feature on the Query Design tab of the Ribbon enables you to set the number of records to be returned. The default setting for this feature is *All*.

QUICK REFERENCE SETTING A QUERY SORT ORDER

Task	Procedure
Set a sort order	■ Display the query in Query Design View. ■ Click the Sort row of the query grid for the field on which you want to sort. ■ Select the appropriate sort order for the field.
Set sort orders for multiple fields	■ Display the query in Query Design View and arrange the fields left to right in the order in which you want Access to sort. ■ Click the Sort row of the query grid for the field on which you want to sort. ■ Select the appropriate sort order for the field. ■ Repeat these steps for each additional field that you want to sort.
Limit the number of records returned	■ Display the query in Query Design View. ■ Choose Design→Query Setup→Return menu and select the number of records you want to view.

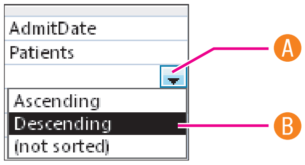
DEVELOP YOUR SKILLS 4.3.1

Set a Query Sort Order

In this exercise, you will set a sort order in the query grid to sort the Daily Admissions results in descending order and limit the results to display the last ten admissions.

Before You Begin: *The Raritan Clinic East Queries database should be open.*

1. **Right-click** the Daily Admissions query and select **Design View** to open the query in Query Design View.
2. Follow these steps to set a sort order in the query grid:



A Click the **Sort** row of the query grid for the AdmitDate column.

B Select **Descending** from the Sort drop-down list.

3. Choose **Design→Query Setup→Return** text box on the Ribbon and type **`10`**.
4. Choose **Design→Results→Run** on the Ribbon to run the query.
   Access displays only the records for the last ten admissions to the clinic.
5. **Close** the query, choosing **No** when prompted to save changes to the query.
   If you save changes after setting the number of records to display, Access saves that number as part of the query. In this case, you do not want to display only ten records each time you run the query.

4.4 Performing Calculations in Queries

Video Lesson labyrinthelab.com/videos

So far, the activities in this lesson have introduced the basics of creating, running, sorting, and selecting records based on criteria. As you developed the queries, you used fields available in field lists from database tables. Access also contains features that enable you to use the query grid to create a *calculated field*—a field that contains no data in a table but uses data in other fields to obtain its value.

A calculated field:

- Creates a new field in the query that can be used in a form or report
- Has a name and can be formatted with properties as a regular field
- Enables you to combine values in two text fields, such as FirstName and LastName, into one field
- Can be used to perform mathematical operations such as add, multiply, etc.
- Updates and recalculates each time you run the query

TIP

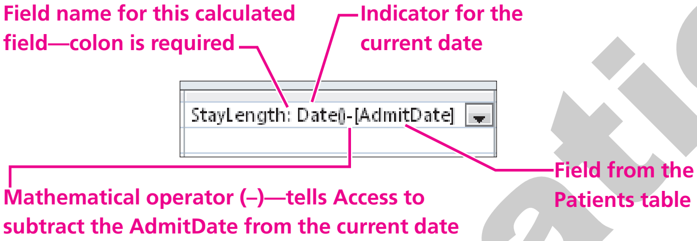
Queries, forms, tables, and reports can contain calculated fields. As a result, it is a good idea to identify calculated fields as you design a database and mark them as calculated fields so that you include in database tables all the fields used in the calculated fields.

Identifying Parts of a Calculated Field

The structure of a calculated field includes a field name and expression elements that tell Access which fields, operators, and punctuation marks to use to create the field. Each calculated field contains the following elements:

ELEMENTS OF CALCULATED FIELDS

Element	Description
Calculated field name	■ The unique name you assign to the field, followed by a colon to separate the field name from the expression.
Field names from existing tables	■ The field containing the data used in the calculation. If field names from different tables appear in the calculated field, Access adds the table name containing the field. Field names appear in brackets [ ].
Arithmetic or comparison operators	■ +, -, /, *, =, >, <, and so forth to compare values or perform mathematical operations. ■ The ampersand (&) is used to join text values from multiple fields such as FirstName & LastName. Required spaces appear within quotation marks (" ").



StayLength: Date()-[AdmitDate]

Field name for this calculated field—colon is required

Indicator for the current date

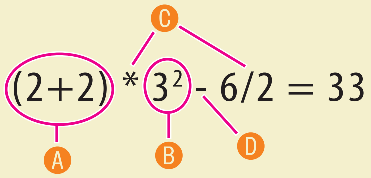
Mathematical operator (–)—tells Access to subtract the AdmitDate from the current date

Field from the Patients table

Identifying Order of Calculations

As you may have learned during your study of math or other computer programs that perform calculations (such as Excel), there is a distinct order used to perform mathematical calculations. Access calculates mathematical operations left to right across a formula by applying the rules of order for calculating. The standard order for performing mathematical operations is often abbreviated *PEMDAS* (you may have learned the phrase *Please excuse my dear Aunt Sally,* a phrase often taught in elementary schools to teach order of operations). The initials represent the order of mathematical operations Access uses, as described in the following Quick Reference table.

QUICK REFERENCE **IDENTIFYING THE ORDER OF MATHEMATICAL CALCULATIONS**



Item	Calculation	Description
A	**P**arentheses	Calculations enclosed in parentheses are performed first.
B	**E**xponentials	Calculations “raised to the power of”, such as squared or cubed, are performed next. Exponents appear as raised numbers, such as in x^2.
C	**M**ultiplication/**D**ivision	Multiplication and division are equal in calculation order and are calculated left to right, after calculations on parentheses and exponentials.
D	**A**ddition/**S**ubtraction	Addition and subtraction are equal in calculation order and are calculated last, left to right across a formula.

Calculating Dates

In addition to performing simple calculations involving dates, Access provides alternative ways to use dates in calculated fields. You can use these expressions to calculate age, number of years in business, and so forth. Samples of dates used in calculated fields are shown in the following Quick Reference table.

QUICK REFERENCE **USING CALCULATED DATES IN EXPRESSIONS**

Sample Field	Returns
CurrentDate: =Date() or CurrentDate: Now()	Displays the current date in the *mm-dd-yyyy* format, where *mm* is the month, *dd* is the day, and *yyyy* is the year.
OrderProcessing: DateDiff("d", [OrderDate], [ShippedDate])	Displays the number of days (d) between the value in the OrderDate field and the ShippedDate field.
(Now()-[DOB])/365	Subtracts the value in the DOB (date of birth) field from the current date and divides the difference by 365 to display the calculated value in years.

Creating and Formatting a Calculated Field

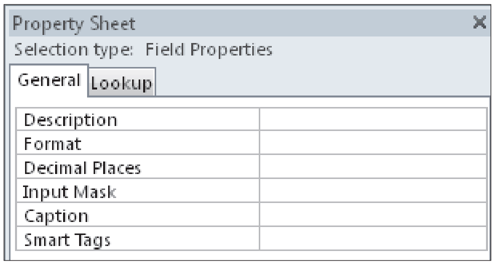
Each calculated field stored in a query appears in a separate column in the query grid. You can type the calculated field into the Field row of the column or create one using tools on the Ribbon in the Query Design View to access the expression builder used to build a calculated field. Other tools on the Ribbon enable you to access field properties for fields used in queries.

Setting Calculated Field Properties

When you create tables in Access, you are able to set field properties—captions, input masks, default values, format, and so forth. Calculated fields often need to be formatted using field properties. To assign field properties to calculated fields, you use the Property Sheet.

FROM THE KEYBOARD

Alt+Enter to open the Property Sheet
F4 to open the Property Sheet



Property Sheet
Selection type: Field Properties

General | Lookup

Description	
Format	
Decimal Places	
Input Mask	
Caption	
Smart Tags	

The Property Sheet for formatting query fields.

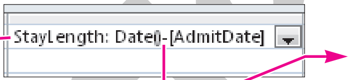
DEVELOP YOUR SKILLS 4.4.1

Create and Format a Calculated Field

In this exercise, you will create a calculated field for a query in your Raritan Clinic East Queries database.

Before You Begin: The Raritan Clinic East Queries database should be open.

1. **Right-click** the Daily Admissions query in the Navigation pane and choose **Design View**.
2. Follow these steps to create a calculated field that calculates the number of days each patient has been in the clinic.



StayLength: Date()-[AdmitDate]

A Click the **Field** row of the next available column in the query grid.

B **Type** the following calculated field exactly as shown here: `StayLength: Date()-[AdmitDate]`

C **Drag** the column border to widen the query grid column so that you can view the entire entry.

3. **Run** the query.
   Access displays the number of days each patient has been in the clinic. Because the current date varies for each student completing this activity, the dates will vary from negative dates to positive dates and may seem as if they are too large. If you were to create such a calculated field in an actively running database, the length of each patient's stay would be much more reasonable. If your query calculated incorrectly, check the field for typographical errors.
4. **Save** changes to the query and **close** it.

Using a Function in a Query Expression

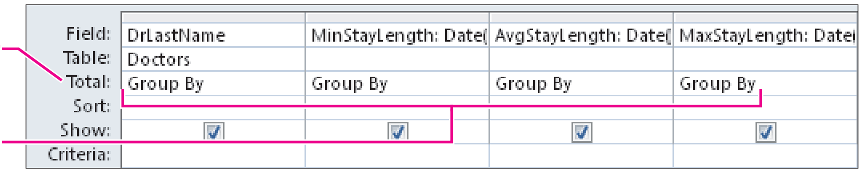
Video Lesson labyrinthelab.com/videos

If you have worked with Microsoft Excel, you are most likely familiar with the types of functions that provide Excel with its calculating power. In Access, you have many of the same types of functions available for performing specific calculations, such as finding the minimum, maximum, and average values, and counting the number of entries in a datasheet. These are known as *aggregate functions* and are built into Access to calculate totals for field values. You can use these functions in queries, forms, and reports to aid in reporting database data.

Adding Functions to the Query Grid

When you want to add functions for totaling, finding averages or similar values based on data contained in database tables, you must first display the Totals row on the query grid. From the totals row, you choose the function you want to calculate for a specified field—using a separate column for each function. For example, if you want to find three different values for the same field—minimum, maximum, and average—you would add the field to the query grid three times—once for each function.

When you first display the Totals row in the query grid, Access places the *Group By* command in the total row for each column. You can use the Group By function to calculate values by each different value in the field. For example, if you added the State field to the grid, you could group by the State field and Access would calculate averages for each state separately.



Field:	DrLastName	MinStayLength: Date(	AvgStayLength: Date(	MaxStayLength: Date(
Table:	Doctors			
Total:	Group By	Group By	Group By	Group By
Sort:				
Show:	☑	☑	☑	☑
Criteria:				

Total row in the query grid.

Group by appears in most columns.

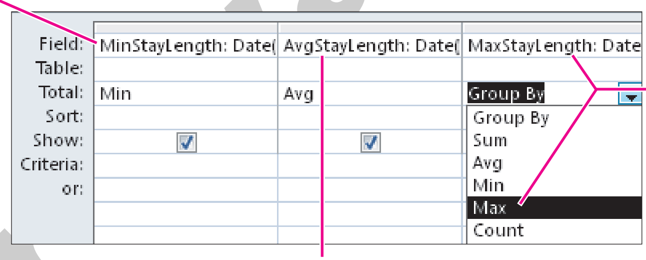
Creating Aliases in Query Fields

Calculated fields are designed to calculate values for summarizing data. As a result, values in calculated fields are not stored in database tables. When you want to use calculated fields for displaying multiple function values in a query, you copy the calculated field to additional columns of the query grid. Then you have to assign a different field name to each additional function value you want to locate.

Alias Example

Suppose you create a calculated field named TotalWages that calculates the weekly wages for each employee. It is conceivable that you may want to find the average, minimum, and maximum values for the calculated field among all employees. When you copy the calculated expression to two additional columns in the query grid, you would need to change the name of the calculated field—the part of the expression that appears before the colon—for each additional instance of the expression. When the same expression is assigned several different field names, the additional field names are referred to as *aliases*.

Original calculated field name finds the Minimum value.



Field:	MinStayLength: Date(	AvgStayLength: Date(	MaxStayLength: Date
Table:			
Total:	Min	Avg	Group By
Sort:			
Show:	☑	☑	
Criteria:			
or:			

Second re-use of the calculated field finds the Maximum.

First re-use of the calculated field finds the Average value.

Identifying Function Types

Simple aggregate functions allow you to count the number of entries in a field, locate the maximum or minimum values in a field, total the values of a group of records, and find the average value from a group of values. Access contains additional functions that enable you to calculate the standard deviation and variance of values. It is also important to know that Access limits or restricts the use of these functions to specific data field types. Some of the more commonly used aggregate functions are identified in the following table.

AGGREGATE FUNCTION TYPES		
Function	**Description**	**Valid Field Data Types**
Sum	Totals values in a field.	Number, Currency
Avg	Averages values in a field.	Number, Date/Time, Currency
Min	Identifies lowest value in a field.	Text, Number, Date/Time, Currency
Max	Identifies highest value in a field.	Text, Number, Date/Time, Currency
Count	Counts the number of values in a field, not counting (blank) values.	All types except multi-value lists
StDev	Calculates standard deviation of the values in a field.	Number, Currency
Var	Calculates variance of the values in a field.	Number, Currency
First	Locates the first record in the group on which you are performing calculations in chronological order without sorting.	All data types
Last	Locates the last record in the group on which you are performing calculations in chronological order without sorting.	All data types

DEVELOP YOUR SKILLS 4.4.2

Use Functions in Queries

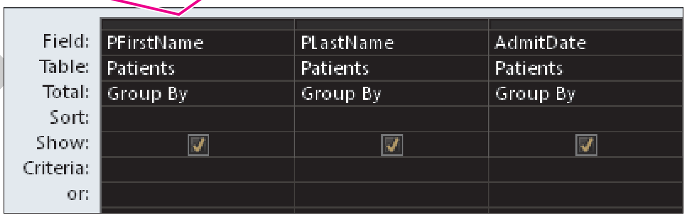
In this exercise, you will create a query by saving an existing query as a new query. Then you will edit the new query so that it displays the query design you need and use functions to identify minimum, maximum, averages, and counts of records for a query in your Raritan Clinic East Queries database.

Before You Begin: The Raritan Clinic East Queries database should be open.

1. **Right-click** the Daily Admissions query in the Navigation pane and choose **Copy**.
2. **Right-click** the Daily Admissions query again, choose Paste, type **`Daily Admissions Summary`** as the new query name, and then click **OK**.
3. Open the Daily Admissions Summary query in Design View and then follow these steps to remove unnecessary fields from the query grid:

A Point to the grey bar above the PFirstName column of the query grid and drag across to DrLastName to select the first three columns.

B Press Delete on the keyboard to remove the fields from the query grid.

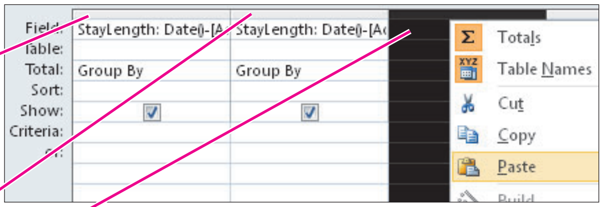


Field:	PFirstName	PLastName	AdmitDate
Table:	Patients	Patients	Patients
Total:	Group By	Group By	Group By
Sort:			
Show:	☑	☑	☑
Criteria:			
or:			

4. Follow these steps to copy and paste the StayLength calculated field to two additional columns in the query grid:

A **Right-click** the grey bar at the top of the StayLength calculated field to select the column and then choose Copy.

B **Right-click** the grey bar at the top of the first blank column in the query grid and choose Paste to paste the calculated field.



Field:	StayLength: Date()-[A	StayLength: Date()-[A(
Table:		
Total:	Group By	Group By
Sort:		
Show:	☑	☑
Criteria:		
or:		

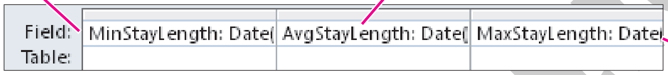
C **Repeat** the procedure outlined in Step B to paste the calculated field in the next column.

You should now have three columns in the query grid that contain the calculated field for StayLength. You need to change the field name for each of the calculated fields.

5. Follow these steps to change the field name for the calculated fields:

A Position the insertion point at the beginning of the first calculated field name and type **Min**.

B Position the insertion point at the beginning of the second calculated field name and type **Avg**.



Field:	MinStayLength: Date(	AvgStayLength: Date(	MaxStayLength: Date
Table:			

C Position the insertion point at the beginning of the third calculated field name and type **Max**.

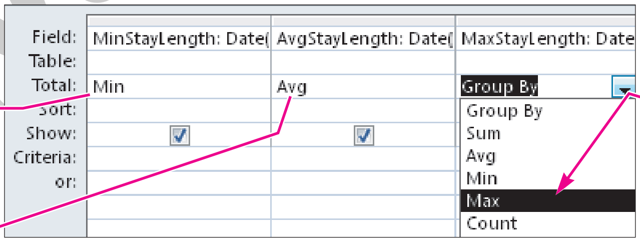
Add Totals and Functions

6. Choose **Design→Show/Hide→Totals** Σ on the Ribbon to display the Total row in the query grid.

7. Follow these steps to choose functions for each of the calculated field columns:

A Click the **Total** row for the first StayLength column in the query grid and choose **Min** from the drop-down list.

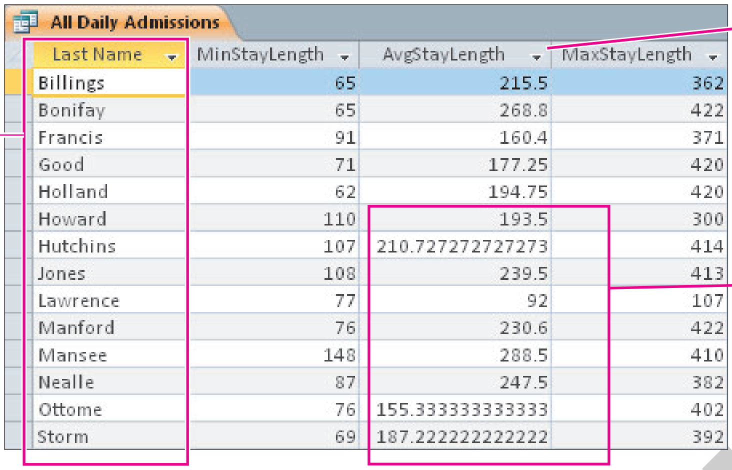
B **Press** Tab to move to the Total row for the next column and choose **Avg** from the drop-down list.



Field:	MinStayLength: Date(	AvgStayLength: Date(	MaxStayLength: Date
Table:			
Total:	Min	Avg	Group By
Sort:			
Show:	☑	☑	
Criteria:			
or:			

C **Press** Tab to move to the Total row for the next column and choose **Max** from the drop-down list.

8. Choose **Design→Results→Run** on the Ribbon to display query results and **adjust** the datasheet column widths to display all data and field names.
Your datasheet should resemble the one shown here:



All Daily Admissions

Last Name	MinStayLength	AvgStayLength	MaxStayLength
Billings	65	215.5	362
Bonifay	65	268.8	422
Francis	91	160.4	371
Good	71	177.25	420
Holland	62	194.75	420
Howard	110	193.5	300
Hutchins	107	210.727272727273	414
Jones	108	239.5	413
Lawrence	77	92	107
Manford	76	230.6	422
Mansee	148	288.5	410
Nealle	87	247.5	382
Ottome	76	155.333333333333	402
Storm	69	187.222222222222	392

Each doctor is listed only one time—grouped by doctor last name.

Each calculated field column shows the column heading that identifies what the value represents.

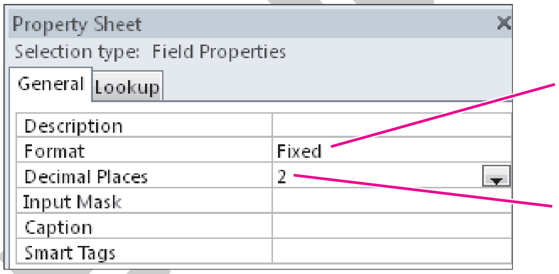
Mixed value formats are quite noticeable.

The number of days you see will differ from those seen here based on the current date on which you run this query.

Format Properties

9. Choose **Home→Views→View** on the Ribbon to switch to Query Design View.

10. Click the **AvgStayLength** field and choose **Design→Show/Hide→Property Sheet** on the Ribbon to open the property sheet and then follow these steps to format the AvgStayLength to display only two decimal points:



Property Sheet
Selection type: Field Properties

Description	
Format	Fixed
Decimal Places	2
Input Mask	
Caption	
Smart Tags	

A Set **Fixed** as the field Format value.

B Set **2** as the number of Decimal Places.

11. **Run** the query again, then **save** and **close** the query.

4.5 Creating Special Types of Queries

Video Lesson labyrinthelab.com/videos

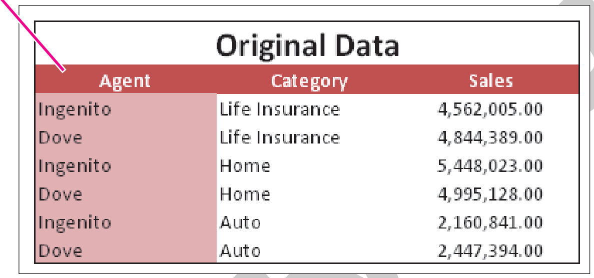
The queries you have created so far are select queries—queries in which Access selects records according to the fields you add to the query grid and the criteria you set. Access also contains tools for creating a number of special types of queries. In this lesson, you will explore three of these special queries:

- Crosstab query
- Find Unmatched query
- Find Duplicates query

Creating a Crosstab Query

Crosstab queries present data by rearranging the layout of fields in the query results datasheet and grouping the data to summarize it. By grouping fields on the left side of the datasheet and arranging other fields across the top, you can calculate sums, averages, counts, or other totals by group and subgroup. Many users find that using crosstab queries makes it easier to analyze data. For example, if you have a database that contains sales records for insurance agents, the category of insurance they sell, and their total sales for each type of insurance, you could create a crosstab query to display the total sales by category for each agent. Such a grouping and summarization might appear as shown in the following illustrations.

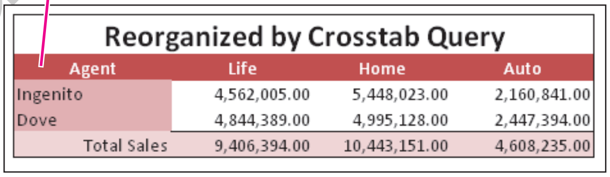
Original data format is arranged by record.



Original Data

Agent	Category	Sales
Ingenito	Life Insurance	4,562,005.00
Dove	Life Insurance	4,844,389.00
Ingenito	Home	5,448,023.00
Dove	Home	4,995,128.00
Ingenito	Auto	2,160,841.00
Dove	Auto	2,447,394.00

Using a crosstab query, you can display the data so that it groups data by agent and category. You can then total the sales for all agents in each category.



Reorganized by Crosstab Query

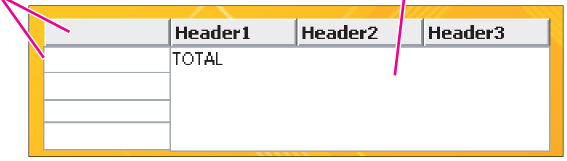
Agent	Life	Home	Auto
Ingenito	4,562,005.00	5,448,023.00	2,160,841.00
Dove	4,844,389.00	4,995,128.00	2,447,394.00
Total Sales	9,406,394.00	10,443,151.00	4,608,235.00

Using the Crosstab Query Wizard

As you work with crosstab queries, you will discover a vast difference between the query grid you have used to create select queries and the crosstab query palette. You can, of course, use the palette to manually construct a crosstab query. Until you become better acquainted with the queries, using the Crosstab Query Wizard is most helpful. Crosstab queries can use both tables and queries as the basis of the query.

Groupings appear down the left column and across the top.

Summarized values appear in the TOTAL area.



	Header1	Header2	Header3
	TOTAL		

The crosstab query palette organizes data so that it is easier to summarize.

QUICK REFERENCE — CREATING CROSSTAB, UNMATCHED, AND DUPLICATES QUERIES USING WIZARDS

Task	Procedure
Create a crosstab query	■ Choose Create→Queries→Query Wizard on the Ribbon. ■ Double-click Crosstab Query Wizard. ■ Follow screens to select objects and fields required to build the query.
Create a find duplicates query	■ Choose Create→Queries→Query Wizard on the Ribbon. ■ Double-click Find Duplicates Query Wizard. ■ Follow screens to select objects and fields required to build the query.
Create a find unmatched query	■ Choose Create→Queries→Query Wizard on the Ribbon. ■ Double-click Find Unmatched Query Wizard. ■ Follow screens to select objects and fields required to build the query.

DEVELOP YOUR SKILLS 4.5.1

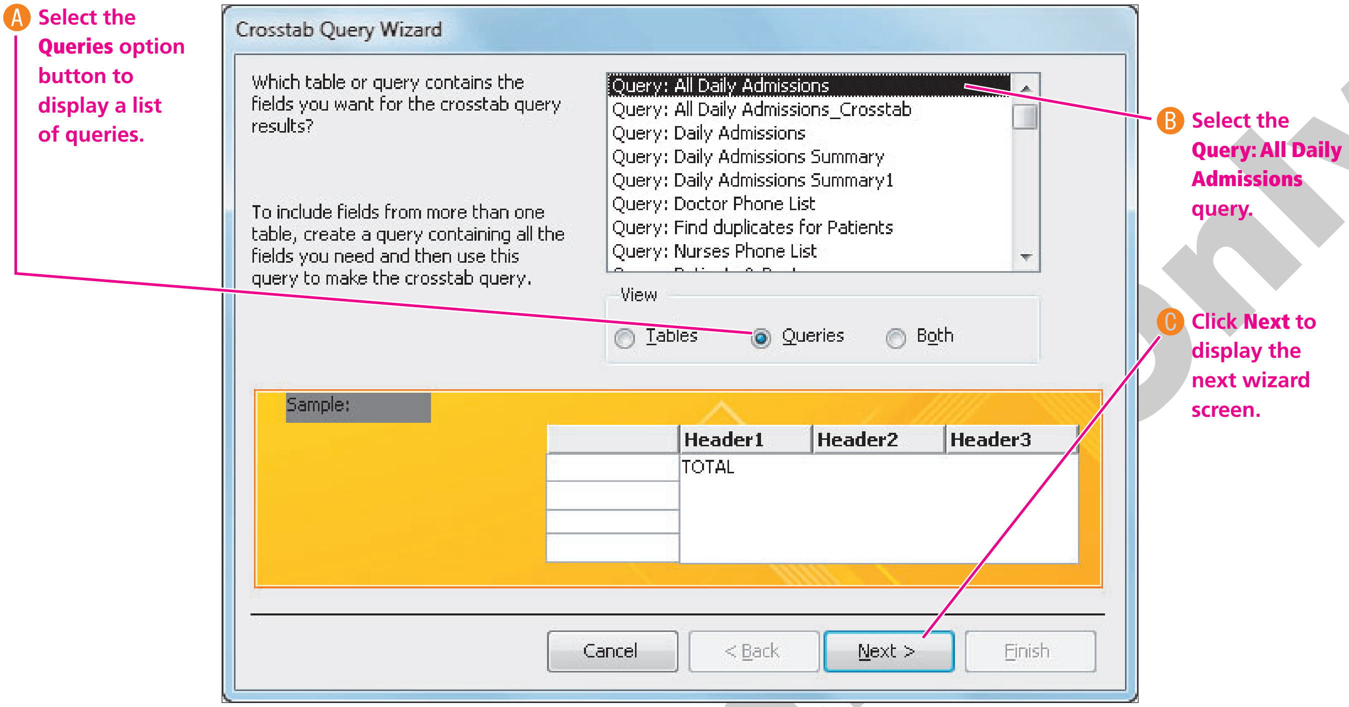
Create a Crosstab Query

In this exercise, you will create a copy of an existing query, modify it, and use the new query to create a crosstab query that lists each doctor and the total number of admissions for each doctor by department.

Before You Begin: The Raritan Clinic East Queries database should be open.

1. **Right-click** the Daily Admissions query and choose Copy; then **right-click** the query again and choose **Paste**.
2. Type **`All Daily Admissions`** as the new query name and click **OK**.
3. **Display** the new query in Design view and add the **Department** field from the Doctors field list to the query design.
4. Save changes to the query and then close it.
5. Choose **Create→Queries→Query Wizard** on the Ribbon to open the New Query dialog box.

6. **Double-click** the Crosstab Query Wizard to launch the Crosstab Query Wizard and follow these steps to select the query to use for the crosstab query:



Crosstab Query Wizard

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

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.

- Query: All Daily Admissions
- Query: All Daily Admissions_Crosstab
- Query: Daily Admissions
- Query: Daily Admissions Summary
- Query: Daily Admissions Summary1
- Query: Doctor Phone List
- Query: Find duplicates for Patients
- Query: Nurses Phone List

View: ○ Tables ● Queries ○ Both

Sample:

	Header1	Header2	Header3
	TOTAL		

A Select the **Queries** option button to display a list of queries.

B Select the **Query: All Daily Admissions** query.

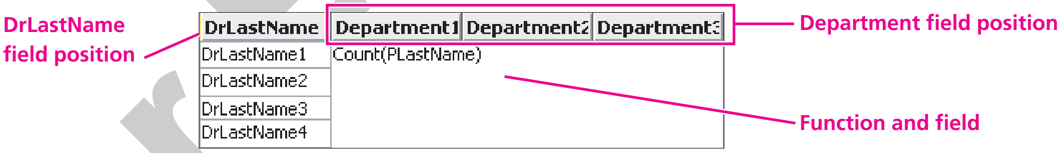
C Click **Next** to display the next wizard screen.

In the second wizard screen, Access wants to know what data you want to display down the left side of the query. In this query, you want the doctor last names to appear down the left column.

7. Choose the **DrLastName** field in the Available Fields list and **move** > the field to the Selected Fields list.

8. Choose **Next**, and then **double-click** the Department field as the field to appear in the column headings.

9. Select **PLastName** in the Fields list and Count in the Functions list to identify the field that contains values and the function you want to calculate.
Your crosstab query grid should resemble the following illustration.



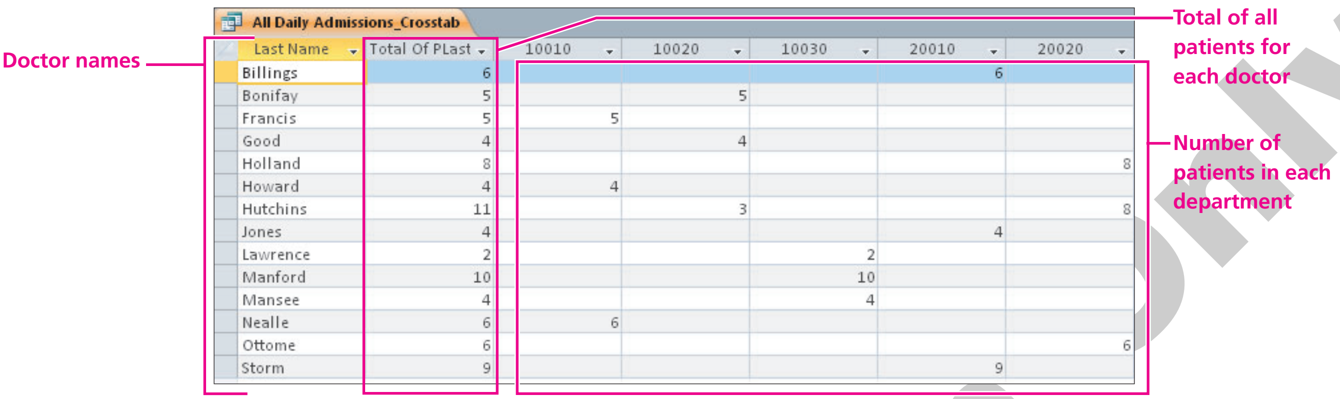
DrLastName	Department1	Department2	Department3
DrLastName1	Count(PLastName)		
DrLastName2			
DrLastName3			
DrLastName4			

DrLastName field position

Department field position

Function and field

10. Click **Next** to display the final page of the Crosstab Query Wizard, and then click **Finish** to accept the defaults Access assigns.
Access runs the query and displays the query results. Your datasheet should resemble the following illustration.


All Daily Admissions_Crosstab
Doctor names
Total of all patients for each doctor
Number of patients in each department

Last Name	Total Of PLast	10010	10020	10030	20010	20020
Billings	6				6	
Bonifay	5		5			
Francis	5	5				
Good	4		4			
Holland	8					8
Howard	4	4				
Hutchins	11		3			8
Jones	4				4	
Lawrence	2			2		
Manford	10			10		
Mansee	4			4		
Nealle	6	6				
Ottome	6					6
Storm	9				9	

11. **Save** and **close** the query.

Naming Special Queries

When you create any of these queries, Access includes the type of query as part of the default query name. You can, of course, modify the query name, but many people choose to include the query type for special queries to make them easier to identify.

Creating Unmatched and Duplicates Queries

Video Lesson labyrinthelab.com/videos

As you should already know, data contained in database tables often shares fields so that you can include data from multiple tables in queries. As a result, it is important that records entered in one table have a matching record in another table. Sometimes, however, you have a table that simply stores data in the database. The data is related to the company and/or the database, but has no official connection to or field in common with other tables in the database.

Access contains two additional query wizards that enable you to create specialized queries for comparing such data—the Find Unmatched Query Wizard and the Find Duplicates Query Wizard.

- **Unmatched Query—**Locates records in one table that have no related records in another table. For example, you could create an Unmatched Query to ensure that each record in an *Orders* table has a corresponding record in the *Customers* table.
- **Find Duplicates Query—**Locates records containing duplicate field values in a single table or query. For example, you could create a Duplicates Query to locate records in the *Orders* table that were entered twice or inquiries submitted more than once.

Identifying the Goal of Unmatched and Duplicates Queries

The goal of each query generated using the Find Unmatched Query Wizard and the Find Duplicates Query Wizard is to identify no records. The queries they generate are simply used to compare entries in database tables. Creating and running these types of queries helps maintain the integrity of the database.

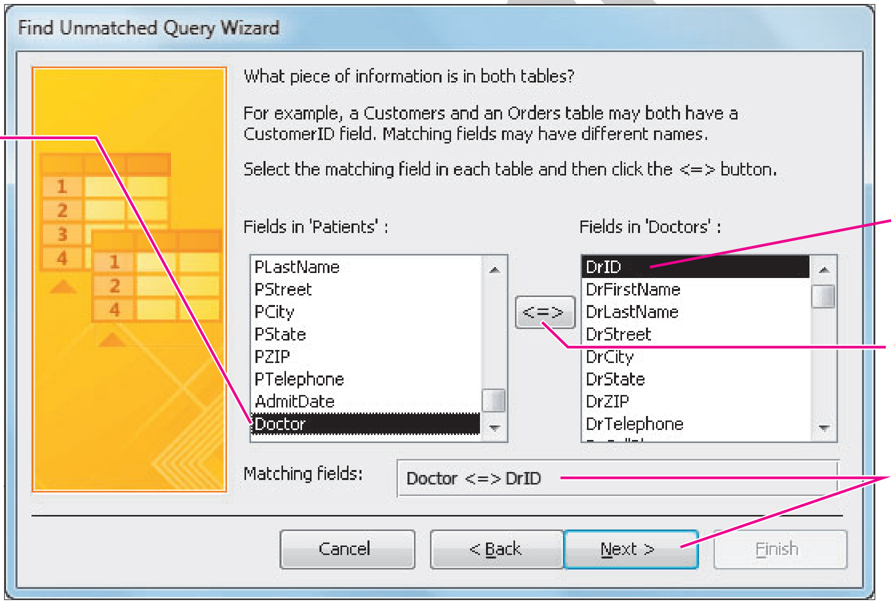
DEVELOP YOUR SKILLS 4.5.2

Create Unmatched and Duplicates Queries

In this exercise, you will use your Raritan Clinic East Queries database to create an unmatched query to locate doctors in the Patients table that have no matching doctor record in the Doctors table. You will then create a duplicates query to identify duplicate patients in the Patients table.

Before You Begin: *The Raritan Clinic East Queries database should be open.*

1. Choose **Create→Queries→Query Wizard** on the Ribbon and double-click Find Unmatched Query Wizard.
   Access presents the first screen in the Find Unmatched Query Wizard. From this screen, you select the table you want to check against another table.
2. **Double-click** Table: Patients to identify the table and automatically advance to the next screen.
3. **Double-click** Table: Doctors to identify the table to compare to the Patients table entries and automatically advance to the next screen.
   The next screen (3) displays a list of fields in both selected tables. From the lists, you will identify the field in the Patients table that must have a matching record in the Doctors table.
4. Follow these steps to identify the fields that should match:



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 'Patients' :
PLastName
PStreet
PCity
PState
PZIP
PTelephone
AdmitDate
Doctor

Fields in 'Doctors' :
DrID
DrFirstName
DrLastName
DrStreet
DrCity
DrState
DrZIP
DrTelephone

Matching fields: Doctor <=> DrID

A Ensure that the Doctor field is selected in the Fields in 'Patients' list.

B Ensure that DrID is the selected field in the Fields in 'Doctors' list.

C Click the **equate** button to indicate that these fields should match.

D Note that the fields selected appear in the Matching Fields box and click **Next.**

Screen 4 asks you to identify the field(s) you want to view in the query results.

5. **Move** [>] the following fields to the Selected Fields list: PLastName, Doctor.
6. Click **Next**, and then click **Finish** to accept the default query name Access assigns.
   The query results datasheet should show no records, the goal of the query. These results ensure that you have a record in the Doctors table for every doctor identified for patients.

Create Duplicates Query

7. **Close** the query, then choose **Create→Queries→Query Wizard** on the Ribbon again.

8. **Double-click** the Find Duplicates Query Wizard.

9. **Double-click** Table: Patients as the table you want to check for duplicates and to automatically advance to the next screen.

10. **Move** > the following fields from the Available Fields list to the Selected Fields list, and then click Next: PatientNumber, PLastName, AdmitDate.
    Access now wants to know what additional fields to display in the query results.

11. Click **Next** without moving any additional fields to the Additional Query Fields list, and then click **Finish**.
    The query results datasheet again shows no entries—the goal of the query.

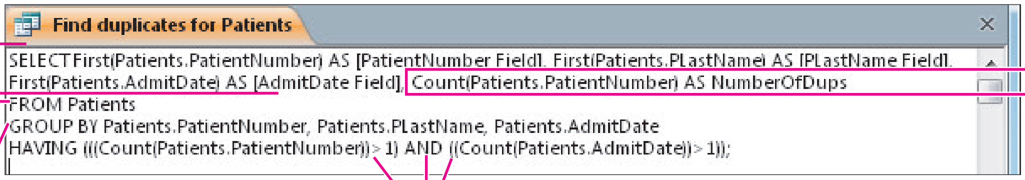
12. **Close** the query and the database, and then **exit** Access.

Viewing Structured Query Language (SQL)

Video Lesson labyrinthelab.com/videos

When you build queries in Access, Access creates the code that contains instructions for displaying the fields and records the query requests according to the criteria you set. In early database programs, those who designed and built the database had to construct the code for displaying the query results. Because Access now does the work for you, it is sometimes interesting to view the code that Access creates for the queries you design. Identifying the instructions Access creates in the code would, possibly, also help you identify reasons why a query might display inaccurate or unexpected results.

In Access, viewing the code is as simple as changing the query view to SQL View. By opening or running a query and changing the view to SQL View, Access reveals the SQL code required to run the query. Study the following illustration of the SQL View for the Find Duplicates query you created in the last exercise.



```
SELECT First(Patients.PatientNumber) AS [PatientNumber Field], First(Patients.PLastName) AS [PLastName Field],
First(Patients.AdmitDate) AS [AdmitDate Field], Count(Patients.PatientNumber) AS NumberOfDups
FROM Patients
GROUP BY Patients.PatientNumber, Patients.PLastName, Patients.AdmitDate
HAVING (((Count(Patients.PatientNumber))>1) AND ((Count(Patients.AdmitDate))>1));
```

The first line identifies the tables and fields from the table that are used in the query. It is preceded by the word *SELECT.*

The last part of the SELECT line identifies what action is to occur—in this case, counting patients.

The FROM line identifies the table containing the fields.

The arrangement of the fields in the display appears in the GROUP line.

Notice the AND condition set by the query and the multiple parentheses used to organize the results datasheet.

4.6 Concepts Review

Concepts Review labyrinthelab.com/acc10

To check your knowledge of the key concepts introduced in this lesson, complete the Concepts Review quiz by going to the URL listed above. If your classroom is using Labyrinth eLab, you may complete the Concepts Review quiz from within your eLab course.

Reinforce Your Skills

REINFORCE YOUR SKILLS 4.1

Create a Query

The First Perk database now contains several tables—the Coffee Types, First Perk Workers and the Menu Items. In this exercise, you will add a query to the database using the First Perk Workers table so that you have a quick reference telephone list of the workers.

1. Launch **Access**, **open** rs-First Perk from the Lesson 04 folder, and **save** it as a new database named **`rs-First Perk Queries`**.
2. Select the **First Perk Workers** table and choose **Create→Queries→Query Wizard** to create a query for the table.
3. Select **Simple Query Wizard** and click **OK**.
4. **Move** the FirstName, LastName, and Telephone fields from the Available Fields list to the Selected Fields list.
5. Choose **Next** and then choose **Finish** to accept the default query name and to open the query.
6. **Print** a copy of the query results datasheet.
7. **Close** the query; leave the database open for the next exercise..

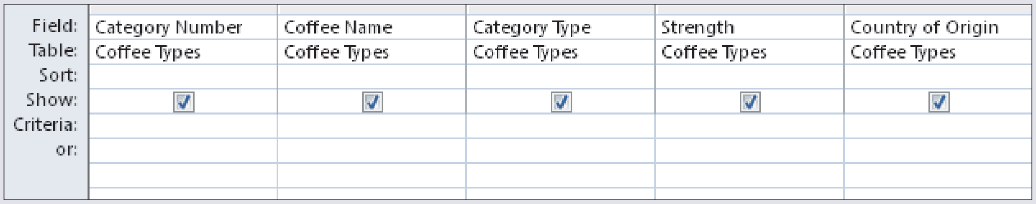
REINFORCE YOUR SKILLS 4.2

Create Queries in Design View

Your database is building nicely. In this exercise, you will create a query based on data contained in the Coffee Types table.

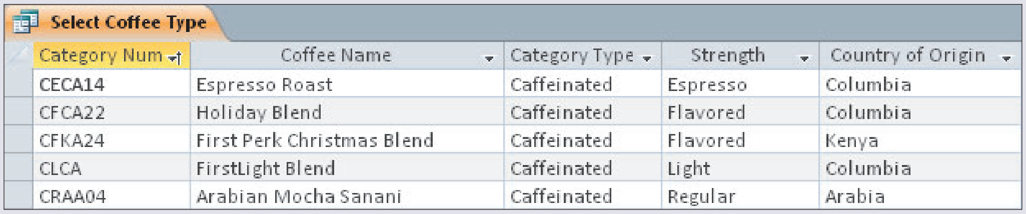
Before You Begin: *The rs-First Perk Queries database should be open.*

1. Select the **Coffee Types** table, and then choose **Create→Queries→Query Design** on the Ribbon.
2. **Double-click** the Coffee Types table in the Show Table dialog box to add the table to the query grid.
3. **Close** the Show Table dialog box.
4. Starting with the Category Number field in the Coffee Types field list, **double-click** each field name (except the Description field) to add it to the query in the order shown.
   When all fields are added, the query grid should match the following illustration.



Field:	Category Number	Coffee Name	Category Type	Strength	Country of Origin
Table:	Coffee Types	Coffee Types	Coffee Types	Coffee Types	Coffee Types
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

5. **Run** the query and review the results.
   Your query results datasheet should appear similar to the following illustration.



Select Coffee Type

Category Num	Coffee Name	Category Type	Strength	Country of Origin
CECA14	Espresso Roast	Caffeinated	Espresso	Columbia
CFCA22	Holiday Blend	Caffeinated	Flavored	Columbia
CFKA24	First Perk Christmas Blend	Caffeinated	Flavored	Kenya
CLCA	FirstLight Blend	Caffeinated	Light	Columbia
CRAA04	Arabian Mocha Sanani	Caffeinated	Regular	Arabia

6. **Save** the query using the query name **`Select Coffee Type`**.

REINFORCE YOUR SKILLS 4.3

Set Query Criteria

One of the tasks of the First Perk database will have to perform is locating coffee according to different criteria. For example, they may want to locate coffee from a specific country or of a specific type. In this exercise, you will set criteria using the Select Coffee Type query to test it and ensure that it is working properly.

Before You Begin: *The rs-First Perk Queries database should be open.*

1. Open the **Select Coffee Type** query in your rs-First Perk Queries database.
2. Choose **Home→Views→View** on the Ribbon to switch to Query Design View.
3. Click the **Criteria** row of the query grid for the Country of Origin column and type **`Brazil`**.
4. Choose **Design→Results→Run** on the Ribbon to run the query.
   Access locates only one record in the Coffee Types table.
5. Choose **Home→Views→View** on the Ribbon to switch to Query Design View.
6. **Double-click** *Brazil* in the Criteria row for Country of Origin and type **`Columbia`**.
7. Click the **Criteria** row for the Category Type field and type **`decaffeinated`**.
8. Choose **Design→Results→Run** on the Ribbon to run the query.
   Access locates three records in the Coffee Types table that meet both criteria.
9. **Close** the query. Choose **No** when prompted to save changes.
   Recall that saving the query with criteria set automatically saves the query with the criteria included.

REINFORCE YOUR SKILLS 4.4

Create and Format a Calculated Field

Businesses often need to calculate distribution times for sales to ensure that the time between receiving an order and shipping the order falls within their preferred timeframe. In this exercise, you will create a calculated field that is designed to calculate the number of days required to process orders for a company named TechCache and set field properties for the calculated field.

1. **Open** the rs-TechCache database from the Lesson 04 folder and **save** it as new database named **`rs-TechCache Queries`**.
2. Open the **Customer Order Data** query in the Navigation Pane, review the data, and then switch to **Design View**.
3. Click the **Field** row of the next available column in the query grid.
4. **Type** the following calculated field exactly as shown:
   `ProcessTime: [Shipped Date]-[Order Date]`
5. **Drag** the column border to widen the query grid column to you can view the entire entry.
6. **Save** changes to the query, and then **Run** the query.
   Access displays the number of days required to process each order along with other fields in the query results. If your query calculated incorrectly, check the field for typographical errors.
7. Choose **Home→Views→View** on the Ribbon to return to Query Design View.

Set Calculated Field Properties

8. Select the **calculated field** and choose **Design→Show/Hide→Property Sheet** on the Ribbon to open the Property Sheet pane on the right side of the query grid.
9. Click the **Caption** property box and type **`Days to Process`** as the caption.
10. Click the **Input Mask** property box and type **`00" Days"`**.
    Notice that the space appears after the opening quotations to ensure there is enough space between the number and the word.
11. Choose **Design→Results→Run** on the Ribbon to run the query again and review the results.
12. **Save** changes to the query and **close** it.

REINFORCE YOUR SKILLS 4.5

Use Functions in Queries

In this exercise, you will create a query by saving an existing query as a new query. Then you will edit the new query so that it displays the query design you need and use functions to identify minimum, maximum, averages, and counts of records for a query in your sb-TechCache Queries database.

Before You Begin: The rs-TechCache Queries database should be open.

1. **Right-click** the Customer Order Dates query in the Navigation pane and choose **Design View**.
2. Choose **File→Save Object As** to open the Save As dialog box, type **`Order Processing Averages`** as the new query name, and then click **OK**.
3. Click the **grey column selection bar** for the following fields in the query grid and **press** Delete to remove them from the query: Order ID, CustNumber, Order Date, Shipped Date.
4. **Right-click** the grey bar at the top of the ProcessTime calculated field to select the column and then choose **Copy**.
5. **Right-click** the grey bar at the top of the first blank column in the query grid and choose **Paste** to paste the calculated field.
6. **Repeat** the procedure outlined in step 5 to paste the calculated field in the next column.
7. **Double-click** the first instance of ProcessTime and type **`MinProcessTime`**.
8. **Double-click** ProcessTime in the second ProcessTime column and type **`MaxProcessTime`**.
9. **Double-click** ProcessTime in the third ProcessTime column and type **`AvgProcessTime`**.

Add Totals and Functions

10. Choose **Design→Show/Hide→Totals** Σ on the Ribbon to display the Total row in the query grid.
11. Set the function on the Total row for each calculated field to match the **Min, Max, and Avg** function identified by the calculated field name.
12. Choose **Design→Results→Run** ! on the Ribbon to display query results.

Format Properties

13. Choose **Home→Views→View** on the Ribbon to switch to Query Design View.
14. Click the **MinProcessTime** field and choose **Design→Show/Hide→Property Sheet** on the Ribbon to open the property sheet and then click the fields shown below in the query grid and change the property values to those shown for each field:

Field	Caption	Decimal Places	Format
MinProcessTime	Minimum Process Time	2	Fixed
MaxProcessTime	Maximum Process Time	2	Fixed
AvgProcessTime	Average Process Time	2	Fixed

15. **Run** ! the query again, then **save** 💾 and close ✕ the query.

REINFORCE YOUR SKILLS 4.6

Create a Crosstab Query

In this exercise, you will create a crosstab query that lists each supplier and identifies the number of different items TechCache purchases from each supplier.

Before You Begin: *The rs-TechCache Queries database should be open.*

1. Choose **Create→Other→Query Wizard** on the Ribbon to open the New Query dialog box.
2. **Double-click** the Crosstab Query Wizard to launch the Crosstab Query Wizard.
3. **Double-click** the Table: Inventory item in the first wizard screen to select the table and automatically advance to the next screen.
4. Choose the **Supplier #.Value** field in the Available Fields list and **move** > the field to the Selected Fields list.
5. Choose **Next**, and then **double-click** the Item Type field as the field to appear in the column headings.
6. Select **Inventory Number** in the Fields list and **Count** in the Functions list to identify the field that contains values and the function you want to calculate.
7. Click **Next** to display the final page of the Crosstab Query Wizard, and then click **Finish** to accept the defaults Access assigns.

Edit the Crosstab Query in Query Design View

8. Choose **Home→Views→View** on the Ribbon to display the Inventory_Crosstab query in Query Design View.
9. Choose **Design→Query Setup→Show Table** on the Ribbon and **double-click** the Suppliers table in the Show Table dialog box to add the table to the query.
10. **Close** the Show Table dialog box.
11. Add the **Supplier#** field in the Suppliers field list to the query grid, placing it in the second column of the grid. where the Item Type field currently appears
    Notice that the Item Type, Inventory Number, and Total of Inventory fields move one column to the right to make room for the new field.
12. Click the **Crosstab** row of the query grid for the Supplier # field, click the **drop-down arrow**, and select **Row Heading** from the list.
13. **Run** ! the query again, view the display, and then **save** and **close** ✕ the query.

REINFORCE YOUR SKILLS 4.7

Create Unmatched and Duplicates Queries

In this exercise, you will use your rs-TechCache Queries database to create an unmatched query to locate orders in the Orders table that have no matching customer record in the Customers table. You will then create a duplicates query to identify duplicate orders in the Orders table.

Before You Begin: *The rs-TechCache Queries database should be open.*

1. Choose **Create→Queries→Query Wizard** on the Ribbon and **double-click** Find Unmatched Query Wizard.
2. **Double-click** Table: Orders to identify the table and automatically advance to the next screen.
3. **Double-click** Table: Customers to identify the table to compare to the Orders table entries and automatically advance to the next screen.
4. Choose the **CustNumber** in the Fields in the ‘Orders’ list.
5. Choose the **CustNumber** in the Fields in the ‘Customers’ list.
6. Click the **equate** button to indicate that these fields should match and click Next.
7. **Move** > the following fields to the Selected Fields list: Order ID, CustNumber, Order Date.
8. Click **Next**, and then click **Finish** to accept the default query name Access assigns.

Create Duplicates Query

9. **Close** ☒ the query, then choose **Create→Queries→Query Wizard** on the Ribbon again.
10. **Double-click** the Find Duplicates Query Wizard.
11. **Double-click** Table: Orders as the table you want to check for duplicates and to automatically advance to the next screen.
12. **Move** > the following fields from the Available Fields list to the Selected Fields list, and then click **Next:** Order ID and Order Date.
13. Click **Next** without moving any additional fields to the Additional Query Fields list, and then click Finish.
14. **Close** ☒ the query and the database, and then **exit** Access.

Apply Your Skills

APPLY YOUR SKILLS 4.1

Create Queries Containing Criteria

Cruises provide a wealth of entertainment and adventure for any special vacation. The Cruises table in the as-Keepsake Cruises database contains data that travel agents might use to locate cruises for travelers inquiring about taking a cruise. In this exercise, you will create queries that you will use to locate records for two of the more popular destinations—Hawaii and the Caribbean.

1. **Open** the as-Keepsake Cruises database from the Lesson 04 folder and **save** it as a new database named **`as-Keepsake Cruises Queries`**.
2. Select the **Cruises** table in the Navigation Pane.
3. **Create** a new query using Query Design View, and add the **Cruises** table to the query grid.
4. **Add** each table field to a separate column in the query design grid and run the query.
5. Switch back to **Design View**, and enter criteria to locate all cruises to **`Hawaii`**.
6. **Save** 💾 the query using the query name **`Hawaiian Cruises`** and run the query.
7. **Print** a copy of the query results window and then **close** ☒ it.
8. **Copy** the Hawaiian Cruise query and create a new query named **`Caribbean Cruises`**.
9. **Edit** the criteria for the Caribbean Cruises query to locate all cruises to the Caribbean.
10. **Save** 💾 changes to the query and **run** it.
11. **Print** a copy of the query results datasheet and then **close** ☒ the query.

APPLY YOUR SKILLS 4.2

Create a Query Using Comparison Criteria

Another common factor in planning a cruise is, of course, cost of the cruise. In this exercise, you will create a query that identifies the sailing dates of cruises that cost less than $1,000. In addition, you will create a second query that identifies cruises sailing during the winter holidays.

Before You Begin: *The as-Keepsake Cruises Queries database should be open.*

1. **Select** the Cruises table in the Navigation Pane.
2. **Create** a new query in Query Design View and add the Cruises table to the query grid.
3. **Add** all the fields to the query grid except the Ship field and save 💾 the query using the query name **`Affordable Cruises`**.
4. **Set** criteria in the query grid to locate all cruises for which an outside cabin costs less than $1,000.
5. **Save** 💾 changes to the query, **run** the query, and **print** a copy of the query results datasheet.
6. **Copy** the Affordable Cruises query to create a new query named **`Holiday Cruises`**.
7. **Display** the query in Design View and set the following criteria in the Date field: **`Between 12/1/10 and 1/10/11`**.
8. **Save** 💾 changes to the query, **run** it, and **print** a copy of the query results datasheet.
9. **Close** all database objects.

APPLY YOUR SKILLS 4.3

Create a Query

In this exercise, you will create a new query using fields from multiple tables.

1. **Open** the as-First Perk database from the Lesson 04 folder and **save** the database using the name **`as-First Perk Queries`**.
2. **Create** a new query named **`Processed Orders`** and add the Orders and Menu items tables to the query.
3. **Add** the following fields from the associated tables to the query grid:

Table	Field
Orders	Items
Menu Items	Price

4. **Run** ❗ the query and **print** a copy of the query results datasheet.
5. **Save** 💾 and **close** ☒ the query.

APPLY YOUR SKILLS 4.4

Create and Format a Calculated Field

The basics for tallying an order using a query are set. In this exercise, you will create a calculated field and format fields in the query. You will continue to use your as-First Perk Queries database in this exercise.

1. Display the **Processed Orders** query in Query Design View and add the **Number** field from the Orders table to the query.
2. Create a **calculated field** in the next available column of the query grid that calculates the total of each item ordered. Use the following values:
   - Field Name: OrderTotal
   - Field from Field Lists: Number, Price
   - Operand: Multiplication (*)
3. **Format** each field in the query grid by setting the following properties for the field:

Field	Caption	Format
Items	Items	
Price	Price	Currency
Number	Quantity	
OrderTotal	Total	Currency

4. **Save** changes to the query.
5. **Run** the query and **print** a copy of the query results datasheet.
6. **Close** the query.

APPLY YOUR SKILLS 4.5

Total Orders Using Functions

Now that the total for each item appears in the query, you can total the value of each order using functions. In this exercise, you will copy a query and save it as a new query, add functions to a query that calculate the total of each order, and use an alias.

Before You Begin: *Your as-First Perk Queries database should be open.*

1. **Copy** the Processed Orders query and create a new query named **`Order Totals`**.
2. Display the **Order Totals** query in Query Design view and **remove** the following fields from the query grid: Items, Price, and Number.
3. Add the **Order ID** field from the Orders table to the query grid and position it to the left of the OrderTotal field.
4. **Copy** the OrderTotal calculated field column and **paste** it into the next available column in the query grid.

5. Change the new **OrderTotal** field name to `OT-1`.
6. Display the **Totals** row in the query grid and set the OT-1 column total to Sum.
7. Delete the **OrderTotal** column and then **run** the query.
8. **Print** a copy of the query results datasheet and then **save** and close the query.

APPLY YOUR SKILLS 4.6

Set Criteria in Queries

Each person who orders receives a copy of their order. In this exercise, you will create a new query that is designed to print only the items included on one order. You will continue to use your as-First Perk Queries database in this exercise.

1. **Copy** the Order Totals query and create a new query named `Order 1001`.
2. Display the **Order 1001** query in Query Design View and set criteria in the query that limits the display to only the items for order **1001**.
3. Add the items field to the grid and place it between the other two fields.
4. **Save** changes to the query and then **run** the query.
5. **Print** a copy of the query results datasheet and then **close** the query.

APPLY YOUR SKILLS 4.7

Create a Crosstab Query

Each order placed at First Perk notes the salesperson who served the customer. In this exercise, you will create a crosstab query that displays the total sales for each employee using your as-First Perk Queries database.

1. Display the **Processed Orders** query in Query Design View and add the SalesPerson and Item Name fields to the query grid.
2. **Save** changes to the query and **close** it.
3. Use the Crosstab Query Wizard to **create** a new crosstab query using the Processed Orders query.
4. Choose appropriate settings from Wizard screens to:
   - **Display** the SalesPerson on the left and ItemName across the top.
   - **Sum** the OrderTotal value to calculate total sales by sales person.
   - **Name** the query using the default name Access assigns.
5. **Display** the new query in Query Design View and **format** the properties for each column in the query grid so that it displays appropriately.

TIP

The values displayed in the Crosstab Query represent sales and money values. Captions and column Format properties can be set to display the values and identify the columns.

6. **Save** changes to the query, **run** it, and **print** a copy of the results.
7. **Close** the query.

APPLY YOUR SKILLS 4.8

Create Duplicate and Unmatched Queries

Checking the database now that you have created queries using the database tables is important. In this exercise, you will check the integrity of the data using the Find Duplicates and Find Unmatched Query Wizards for your as-First Perk Queries database.

1. **Close** all database objects that might still be open, and then **run** the Find Duplicates Query Wizard to check the First Perk Workers table for duplicate entries and **display** just the employee number, first name, and last name in the results.
2. **Run** ! the query and **print** a copy of the query results.
3. Run the **Find Unmatched Query Wizard to** create the following two unmatched queries, **printing** a copy of the query results as you run each query:
   - **Compare** the Category # field in the Categories table to the CategoryNumber field in the Menu Items table and display the Description field in the results.
   - **Compare** the Orders.Items.Value field in the Processed Orders query to the Item Number field in the Menu Iterms table and display the OrdersNumber field in the results.
4. **Close** ☒ all open database objects, **close** ☒ the database, and **exit** Access.

Critical Thinking & Work-Readiness Skills

In the course of working through the following Microsoft Office-based Critical Thinking exercises, you will also be utilizing various work-readiness skills, some of which are listed next to each exercise. Go to labyrinthelab.com/workreadiness *to learn more about the work-readiness skills.*

4.1 Create Queries

The Raritan database you used in this lesson has additional queries Raritan managers would like to have. Put yourself in the position of an administrator for one clinic department and identify at least four uses for data contained in database tables. Then, determine how this table data could be retrieved using queries. Create two of these queries in the ct-Raritan Clinic East database (Lesson 04 folder), saving the edited database as **`ct-RCE Queries`**. Run the queries and set properties to improve the appearance of the query results datasheet. Print a copy of the query results datasheets.

WORK-READINESS SKILLS APPLIED

- Serving clients/customers
- Organizing and maintaining information
- Using computers to process information

4.2 Explore Data Online

Raritan Clinic East wants to include a list of local pharmacies in its database so that when patients call in for a prescription renewal, the clinic will be able to quickly locate the phone number to call in the renewal. Go online and locate three pharmacies or drug companies in your area, as well as one mail order prescription service, such as Medco Health. Review their search tools and medications lists. Then, on a clean sheet of paper, develop a list of the fields you would include if you were to build a table for pharmacies. Identify how data from the Pharmacies table might be used to create a query that could be used by all personnel as a quick reference sheet for serving customers when they call for a prescription renewal.

WORK-READINESS SKILLS APPLIED

- Acquiring and evaluating information
- Organizing and maintaining information
- Serving clients/customers

4.3 Get Online Medical Advice

James Elliott has just learned that a number of online services are now offering online medical advice. Such sites enable you to enter a series of symptoms and obtain a preliminary diagnosis that identifies what types of medical issues might cause such symptoms. Locate at least one such site. Explore the site to determine how it works, what sources are used, what search tools are available, and what it costs. If possible, enter a few symptoms (real or imaginary) and see what type of diagnosis it gives. If working in a group, discuss how James might suggest implementing such a service at Raritan Clinic East. If working alone, type your response in a Word document named **`ct-Questions`** saved to your Lesson 04 folder.

WORK-READINESS SKILLS APPLIED

- Showing responsibility
- Selecting technology
- Improving or designing systems