

# Microsoft Access 2016

## Intro to Select Queries



# Access 2016: Intro to Select Queries

2.0 hours

Queries compile data from tables into a desired dataset. You can choose multiple fields from one or more data tables, and set criteria to limit the records displayed. By its simplest definition, a query is a saved filter, but they can combine fields, perform calculations, and create summaries. Topics for this workshop include using data filters, setting custom sorts, using criteria, creating joined field expressions, calculating simple math, summarizing with totals, formatting numbers, and creating crosstabs.

Filters .....	1
Filter with the Filter Arrow .....	1
Filter by Selection .....	1
Queries .....	2
Create a Query in the Design View .....	2
Custom Sort Orders .....	3
Individual Fields .....	3
* Field .....	3
Create Query with Simple Query Wizard .....	4
Using Multiple Criteria for Exact Values .....	4
OR (multiple rows) .....	4
AND (single row) .....	4
ANDs and ORs .....	5
Enter Data Prompt/Missing Parameter Value .....	5
Using Multiple Criteria for a Range of Values .....	6
Join Fields with & .....	6
Expression Builder .....	7
Building Join Statement .....	7
Building Equations .....	8
Using Totals .....	9
Unique Records / No Duplicates .....	9
Summarizing Visit Length by Doctor .....	10
Working with Multiple Tables .....	10
Crosstab Query .....	11
Crosstab Query Wizard .....	11
Create Query with Simple Query Wizard .....	11
Practice: Make another Crosstab Query .....	13



**Pandora Rose Cowart**  
Education/Training Specialist  
UF Health IT Training

C3-013 Communicore  
PO Box 100152  
Gainesville, FL 32610-0152

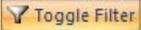
(352) 273-5051  
[prcowart@ufl.edu](mailto:prcowart@ufl.edu)  
<http://training.health.ufl.edu>

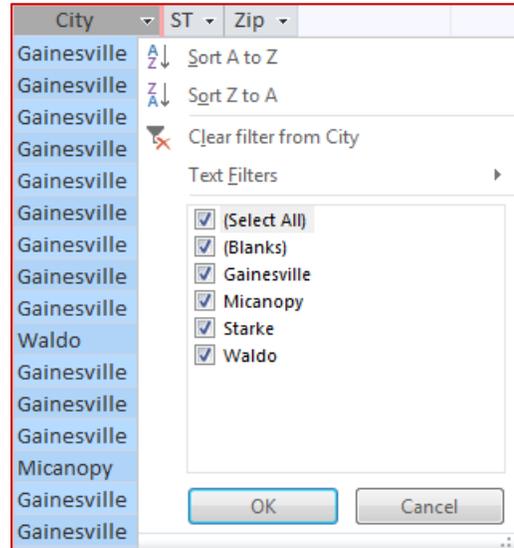
## Filters

The Tables and Queries have filter arrows next to each column heading in the Datasheet view. The filter tools allow you to create a temporary "query" based on criteria you set.

If we did not want to see the people in Gainesville uncheck "Gainesville" and click OK. Access will filter out, hide, the records all of the Gainesville records. You can filter by as many fields as you would like, and Access will continue to limit the displayed dataset based on your filters.

The filtered fields show a small funnel next to in the column heading, and **Filtered** appears next to the navigation buttons at the bottom of the window.

Click on **Filtered** at the bottom of the table to clear the filter. Use this button to toggle between your last filter and all the records, or use the  button on in the **Sort & Filter** section of the **Home** tab.



### **Filter with the Filter Arrow**

- 1. Open Table Patients ..... (76 records)
- 2. Click on Filter Arrow for City field
  - Uncheck Gainesville, OK ..... (21 records)

### **Filter by Selection**

- 1. In the City column, Right-click on a "Micanopy"
  - Choose **Does Not Equal "Micanopy"** ..... (12 records)
- 2. Press the Toggle Filter button on the Home tab ..... (76 records)
- 3. In the Address column, Select and then Right-click on an "Ave"
  - Choose **Contains "Ave"** ..... (13 records)
- 4. Close and Save with only the 13 records showing
- 5. Open Table Patients ..... (76 records)
- 6. Toggle Filter ..... (13 records)

If you save the table, Access will remember the most current filter. When you open the database object, toggle the filter from the home tab, or the navigation area at the bottom of the window.

As soon as you make a new filter, Access will forget about the previous one. To create a 'permanent' filter, create a Query.

---

## Queries

The top half of the query window shows the tables and queries used as your record source. If there are relationships already set, you will see the tables are connected. If not, you can drag and drop from one record source to another.

The bottom half of the Query window shows the fields for the dataset of your query.

To add a field to the bottom half of the screen, you can:

- Click and drag the field name from the Table in the top half of the window to the field row
  - Use the Shift key to select multiple connected fields
  - Use the Ctrl key to select/unselect specific field
- Double-click on the field from the table (Access will place it at the end of all the fields)
- Choose the field name from the drop-down list provided
- The \* field at the top of the tables and queries, will select all the fields in the record

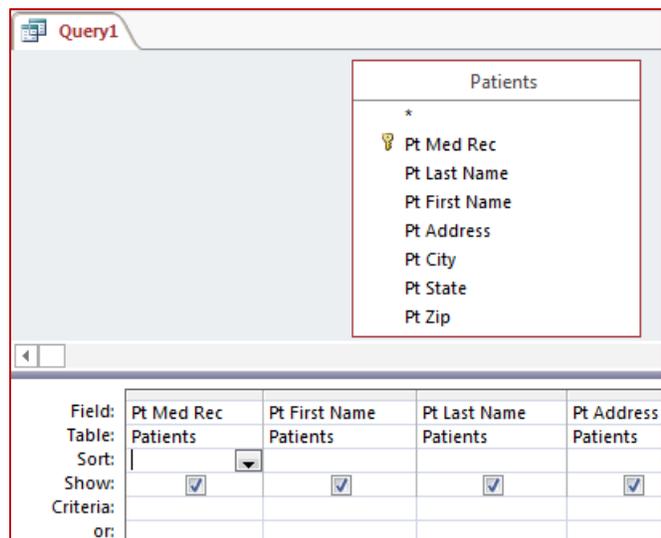
To delete a field from the bottom half of the screen, you can:

- Erase the field name from the field line with the Backspace or Delete key
- Use the Delete Column button in the Design tab
- Right-click on the selection bar at the top of the field and choose Cut

### Create a Query in the Design View



1. From the Create tab choose Query Design
2. Add the Table Patients and close the Show Table window
3. Add the fields below using different methods
  - Double-click *Pt Med Rec #*
  - Double-click *Pt Last Name*
  - Drag and Drop *Pt First Name* onto the *Pt Last Name* in the bottom of the Query, this should put it before Pt Last Name
  - Change next blank column/field to Pt Address from drop down list
  - Click on Pt City, Ctrl-click on Pt Zip
    - ~ This will pick up Pt City and Pt Zip
    - ~ Click in an empty space next to table to drop selection
  - Click on City, shift click on Zip
    - ~ This will pick up Pt City, Pt State, and Pt Zip
  - Drag all three to next blank



---

## Custom Sort Orders

### Individual Fields

By default, Access sorts the dataset by the Primary Key field.



1. View the Datasheet
2. Set a Sort Order
  - Sort Ascending by Pt Last Name
  - Sort Descending by Pt City

3. Clear Sort

4. In the Design View, set a Custom Sort Order

- Sort by Pt Last Name Ascending
- Sort by Pt City Descending

Field:	Pt City	Pt Med Rec	Pt First Name	Pt Last Name
Table:	Patients	Patients	Patients	Patients
Sort:	Descending			Ascending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

~ If you double-click in the Sort row, Access will toggle through the options

5. View the Datasheet

- When we run the query, Access sorts by the Pt Last Name and then the Pt City

6. In the Design view, move the Pt City field to the front of the fields

7. View the Datasheet

- The data is sorted by Pt City and then Pt Last Name
- If desired, you can move the Pt City in the datasheet view, without changing the design

8. Close and save as **Patients Sort 1**

### \* Field



1. Create Query in Design view, Patients
2. Move \* field down (double-click or drag)
3. Move down Pt City and set the sort to Descending
4. In the Datasheet view, notice two City Fields (Patient.Pt City and Field0)
5. In the Design View, turn off the **Show** option for the Pt City
6. Move down Pt Last Name field and set the sort order to Ascending, turn off the **Show** option
7. Close and save as **Patients Sort 2**

Field:	Pt Med Rec	Pt City	Pt Last Name
Table:	Patients	Patients	Patients
Sort:		Descending	Ascending
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:			
or:			

---

### Create Query with Simple Query Wizard

- 1. From the Create tab choose Query Wizard
- 2. Select the Simple Query Wizard
- 3. Use the table patients and add all the fields (next)
- 4. Name the Query as **Patients Query** and choose **Modify the design view** (finish)
  - Notice brackets around field names
- 5. Close Query
  
- 6. Right-click on Patients Query in the Navigation Pane and Choose Design View
  - Notice brackets are gone
    - ~ The brackets are necessary when we are building expressions
  - Set a Criteria for the Pt City field to be "Waldo"
- 7. Go to the Datasheet view..... ( 9 records)
- 8. Go to the SQL View
  - SQL = Structured Query Language. This is the most common language for building queries within relational databases. We will not work in the code.
  - `SELECT Patients.[Pt Med Rec], Patients.[Pt Last Name], Patients.[Pt First Name], Patients.[Pt Address], Patients.[Pt City], Patients.[Pt State], Patients.[Pt Zip]`  
`FROM Patients WHERE (((Patients.[Pt City])="Waldo"));`

### Using Multiple Criteria for Exact Values

#### OR (multiple rows)

- 1. Set the first line of the Criteria for Pt City to Waldo
- 2. Set the second line of the Criteria (OR) to Starke
- 3. Go to the Datasheet view..... (12 records)
- 4. Clear the criteria lines

CITY	ZIP
Patients	Patients
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
"Gainesville"	"32608"

#### AND (single row)

- 1. In the Design view, set the criteria for Pt Zip to "32608"
- 2. Go to the Datasheet view..... (12 records)
  - 32608 is a Gainesville zip code, but this is showing records for Micanopy
- 3. In the Design view, set the criteria for Pt City to "Gainesville"
  - Stay on the same line as the 32608 criteria under Pt Zip
- 4. Go to the Datasheet view..... ( 9 records)

ANDs and ORs



1. In the Design view, on the OR line, set the criteria for Pt City to "Waldo"
  - Access sees: (Gainesville AND 32608) OR Waldo
2. Go to the Datasheet view..... (18 records)
3. Press the **F12** key to **Save As...**
4. Save as **Patients Criteria - Exact** and close the Query

CITY	ZIP
Patients	Patients
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
"Gainesville"	"32608"
"Waldo"	

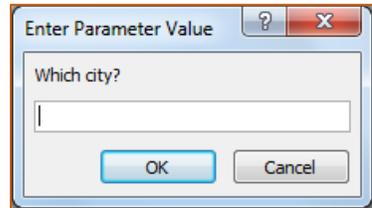
**Enter Data Prompt/Missing Parameter Value**

Instead of changing the criteria every time we want a different set of data, we can set Access to prompt us for criteria by creating a missing parameter value. You will see the Enter Parameter Value window whenever a field is missing, so sometimes this highlights errors, but here we are doing it on purpose.

To create our own field, use the brackets around the text you want to see in the dialog box, be sure not to use a field name that already exists in your data set.



1. Create Query in Design view using the table Patients
2. Double-click on title of the Patients table to select everything in the table except the \*
3. Drag the selection of fields into the bottom of the window
4. Set Criteria for Pt City to [Which city?]
  - Use the brackets
5. Go to the Datasheet view
6. Because there is not a field called **Which city?** Access will open a dialog box asking for the parameter value. Use the Refresh button on the Home tab to pull in a new set. Spelling counts!
  - Gainesville ..... (55 records)
  - Micanopy ..... ( 9 records)
  - Waldo ..... ( 9 records)
  - Starke ..... ( 3 records)
7. Close and Save as **Patients by City**
8. Open Query to test missing parameter value



Field:	CITY
Table:	Patients
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	[Which city?]
or:	

Examples

- [Enter the city you wish to view]
- Between [Enter First Date] and [Enter Last Date]
- Like [First letters of Last Name]&"\*"

---

### Using Multiple Criteria for a Range of Values

- 1. Create Query in Design view using the table Patients
- 2. Double-click on title of the Patients table to select everything in the table except the \*
- 3. Drag the selection of fields into the bottom of the window

Clear each criteria before entering a new one.

- 4. Set the following Criteria for Pt Zip and view the Datasheet
  - >=32600 **OR** <= 32699 ..... (76 records)
  - >=32600 **AND** <= 32699 ..... (51 records)
  - **Between** 32600 **And** 32699 ..... (51 records)
- 5. Wild Cards criteria for Pt Last Name: (\* = Multiple Characters; ? = One Character)
  - j\* (Like "j\*") ..... (10 records) Begins with J
  - \*s (Like "\*s") ..... (21 records) Ends with S
  - j\*s (Like "j\*s") ..... ( 6 records) Begins J & Ends S
  - \*e\* (Like "\*e\*") ..... (42 records) Contains E
  - Like "?e\*" ..... (11 records) 2nd char is an E
- 6. Set Criteria for Pt Address
  - Like \*box\* ..... (14 records)
  - Like \*box \* ..... (12 records) with space
  - Not Like \*box \* ..... (64 records)
- 7. Set all three of the criteria below
  - Pt Address: Not Like \*box \*
  - Pt Zip: Between 32600 and 32699
  - Pt Last Name: Like \*e\* ..... (21 records)
- 8. Save as **Patients Criteria - Range**

### Join Fields with &

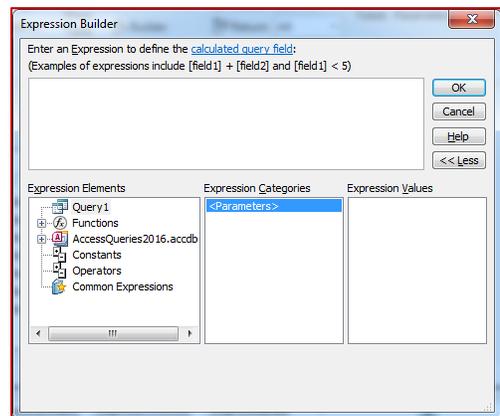
- 1. Create Query in Design view, using the table Patients
- 2. In the first blank field blank type: [Pt First Name]&[Pt Last Name]
- 3. View Datasheet view
  - Title of column is Expr1 (expression)
  - No Spacing between the fields
- 4. In the Design view, right-click on expression and choose ZOOM
  - If desired, click on the **Font** button to make the text larger
  - Change the expression to **Patient Name: [Pt First Name] & " " & [Pt Last Name]**
- 5. Go to the Datasheet view
- 6. Close and save as **Patient Names**

## Expression Builder

Access has an Expression Builder to help you create your joins and calculations. To use this tool you can right-click in the field box, in the Design view of a Query, and choose Build...

You can find built in functions in this builder. In the left section, Expression Elements, double-click on **Functions**, and open the list of Built-In Functions. These include some statements you may be familiar with from Excel: Sum, Count, Max. Here Average() is Avg, and If() is iif. In Access, we use our field names instead of ranges of cells.

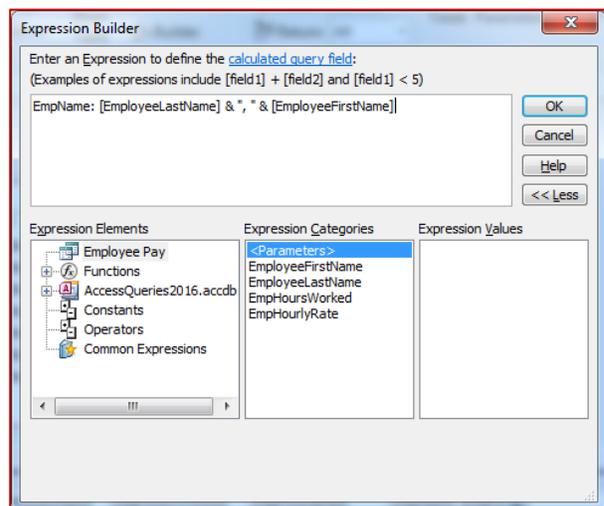
- Sample Derived Expression
  - LastFour: Right([SSN],4) # Find last four characters of the SSN
- Sample Logic Expression
  - Taxes: iif([Exempt]=Yes, 0, [Price]\*0.06) # Calculate the taxes on nonexempt items
- Sample Calculated Expression
  - Total: [AmtSold] \* [TotalPurchased]



## Building Join Statement



1. Create Query in Design View, using the table Employees
2. Add the following Fields:
  - EmployeeFirstName
  - EmployeeLastName
  - EmpHoursWorked
  - EmpHourlyRate
3. Right-click next blank column, and choose **Build**
  - Expression Categories section is blank, click **Cancel**
4. Save Query as **Employee Pay**
5. Try Expression Builder again
  - Double-click EmployeeLastName
  - Type: "&", "&"
  - Double-click EmployeeFirstName
6. Click OK and View Datasheet
7. In the Design view, rename Expr1 to EmpName and move this expression to the first column
8. Delete the **EmployeeFirstName** and **EmployeeLastName** fields from bottom of the Query Design



EmpName: [EmployeeLastName] & ", " & [EmployeeFirstName]	EmpHoursWorked	EmpHourlyRate
	Employees	Employees

---

### Building Equations

#### Calculate **Gross Pay**

- 1. Right-click next blank column, choose BUILD
  - Type - **EmpGrossPay**:
  - Double-click to get **[EmpHoursWorked]**
  - Double-click to get **[EmpHourlyRate]**
  - Erase first <<expr>>, change second one to \* (to multiply)
- 2. Expression should read - **EmpGrossPay: [EmpHoursWorked] \* [EmpHourlyRate]**
- 3. View Datasheet to see the results

#### Calculate **Deductions**

- 1. In the next blank column type -- Deductions:0.175
  - Change Format properties to **Percent** (right-click to get property window)
- 2. View the Datasheet, we now have a constant in the Deduction column
- 3. Use Expression Builder to modify the equation to show the deduction amount instead of the deduction rate; don't forget to save first.
- 4. Change the expression to **Deductions: 0.175\*[GrossPay]**
- 5. Right-click Properties, Format -> Currency

#### Calculate **NetPay**

- 1. NetPay: **[GrossPay] -[Deductions]**
- 2. View Datasheet
- 3. Close and Save query

GrossPay: [EmpHoursWorked]*[EmpHourlyRate]	Deductions: 0.175*[grosspay]	NetPay: [GrossPay]-[Deductions]

EmpName	EmpHoursWorked	EmpHourlyRate	GrossPay	Deductions	NetPay
Appleton, Annie	40	\$10.00	\$400.00	\$70.00	\$330.00
Blackthorne, Billie	40	\$15.00	\$600.00	\$105.00	\$495.00
Carson, Carly	40	\$12.50	\$500.00	\$87.50	\$412.50
Dawson, Devon	35	\$7.00	\$245.00	\$42.88	\$202.13
Edwards, Edgar	25	\$6.50	\$162.50	\$28.44	\$134.06

## Using Totals

By default, the **Total** row does not appear in the Design view of a Query. To display this row, click on the **Totals** button on the right side of the Design tab, or right-click in the bottom of the Query and choose **Totals**.

*Without Totals*

Field:	Visit Date
Table:	Visit Info
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	
or:	

Visit Date
4/28/2009
6/17/2009
4/28/2009
4/28/2009
6/17/2009

*With Totals (group by)*

Field:	Visit Date
Table:	Visit Info
Total:	Group By
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	
or:	

Visit Date
4/28/2009
6/17/2009

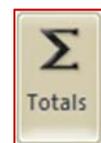
Total functions include:

- **Group By:** Group this field (Hide Duplicate Values)
- **Sum:** Find Sum within each Group
- **Avg:** Find Average within each Group
- **Min:** Find Minimum Value within each Group
- **Max:** Find Maximum Value within each Group
- **Count:** Find Number of Items within each Group
- **StDev:** Find Standard Deviation within each Group
- **Var:** Find Variance of each Group
- **First:** Find First Value entered of each group
- **Last:** Find Last Value entered of each Group
- **Expression:** No Totals, just evaluate expression within the field
- **Where:** Find the dataset based on this criteria, by default this will not show this field

### Unique Records / No Duplicates



1. Create Query in Design view using the table Doctors
2. Add the field **DocFirst**
3. View Datasheet ..... ( 3 records)
4. In the Design view, drag the Table **Visit Info** into the top half of the Query window
  - Because a relationship is set, the tables will already be linked
5. Don't add anything to the query, but view the Datasheet ..... (162 records)
  - There are still only three doctors, but adding the **Visit Info** table, which has 162 records, changes the Query result to show the doctor for each Visit Info record.
6. From the Design tab turn on the **Totals** (look for the sigma on the right side)
7. View Datasheet ..... ( 3 records)
8. In the Design view, add the field **Visit Length**
9. View Datasheet ..... ( 15 records)
  - 3 Doctors, 5 Length options.  $3 * 5 = 15$  records



---

*Summarizing Visit Length by Doctor*

- 1. In the Design view, change the **Visit Length's Totals** row from **Group by** to **Sum**
- 2. View Datasheet ..... ( 3 records)
- 3. In the Design view, add a title to Visit Length
  - **Total Time: Visit Length**
- 4. Add another **Visit Length** field to the bottom of the Query
  - Change the second **Visit Length** field from Group by to **Average**
  - Add the title **Avg Time**
  - Change the **Format** property to **Standard**
    - ~ If needed, right-click on the field to open the Properties
    - ~ Access ignores the number of decimal places until you choose a **Format**
- 5. View Datasheet ..... ( 3 records)
- 6. Close and Save as **Doctor Time Summary**

Field:	DocFirst	Total Time: Length	Avg Time: Length
Table:	Doctors	Visit Info	Visit Info
Total:	Group By	Sum	Avg
Sort:			

DocFirst	Total Time	Avg Time
Sallye	2250	41.67
Samuel	2205	40.83
Sidney	2295	42.50

***Working with Multiple Tables***

- 1. Create Query in Design view using the tables Doctors and Visit Reasons
  - There is no relationship line between the two tables. There is no inherent link between these two datasets. Any fields used here will give you every possible combination of records, even if they don't exist in the actual datasets.
  - When does the doctor treat a reason? When is the reason treated by the Doctor? During the visit. The **Visit Info** table must be part of the Query to get an accurate result on your data.
- 2. In the Design view, drag the Table **Visit Info** into the top half of the Query window
- 3. Add the following Fields:
  - From Table Doctors: **DocFirst**
  - From Table Visit Reasons: **VisitReason**
  - From Table Visit Info: **Visit Length**
- 4. Turn on Totals
- 5. Change the **Visit Length** Total to **Sum**
- 6. View Datasheet ..... ( 25 records)
  - 3 Doctors \* 9 Reasons = 27 records, but we only get 25. This is because at this time, there is only one *Headache* record. As the other Doctors treat Headaches, they will appear in this datasheet.
- 7. Turn off Totals ..... (162 records)
- 8. Close and Save as **Doctors by Reason**

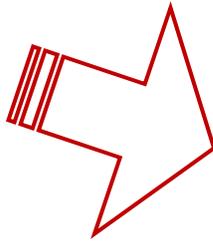
---

## Crosstab Query

Crosstab queries will put the unique values of a field across the top of the Query result as Field Headers, and summarizes a field for each data point. Think of it as a simple version of a PivotTable. We will use the wizard in class, but you can build your own once you become comfortable.

**Example:**

<u>Item</u>	<u>Color</u>	<u>Sold</u>
Blouses	Blue	5
Blouses	Red	10
Pants	Blue	3
Pants	Red	12
Socks	Blue	1
Socks	Red	5



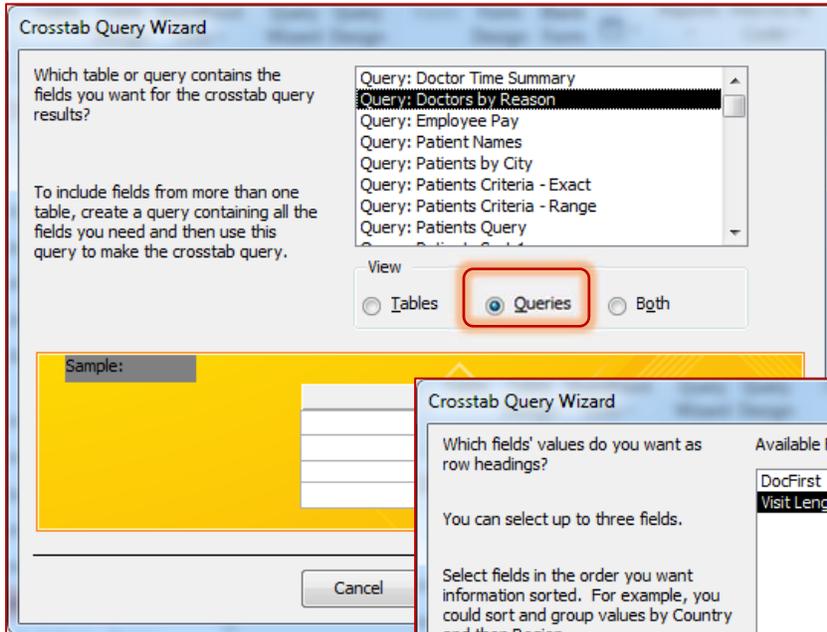
<u>Item</u>	<u>Blue</u>	<u>Red</u>	<u>Total Sold</u>
Blouses	5	10	15
Pants	3	12	15
Socks	1	5	6

## Crosstab Query Wizard

### **Create Query with Simple Query Wizard**



1. From the Create tab choose **Query Wizard**
2. Select the **Crosstab Query Wizard**
3. Since we want information from three different tables, we have to create the query with all the fields we need before we run this wizard. We created this query as **Doctors by Reason**. You won't see this query right away because the wizard is only displaying the tables.
  - Change the **View setting** to show the **Queries**, choose **Query: Doctors by Reason (Next)**
4. You can use up to three fields as your Row Headings. Watch the Preview to see if you're getting the desired fields along the side of the Query result.
  - Select the **VisitReason** as your Row Heading (*Next*)
5. Select the **DocFirst** field as your Column Heading (*Next*)
6. Select the **Sum** option to summarize our **Visit Length**
7. Name the Query: **Doctors by Reason Crosstab Sum**

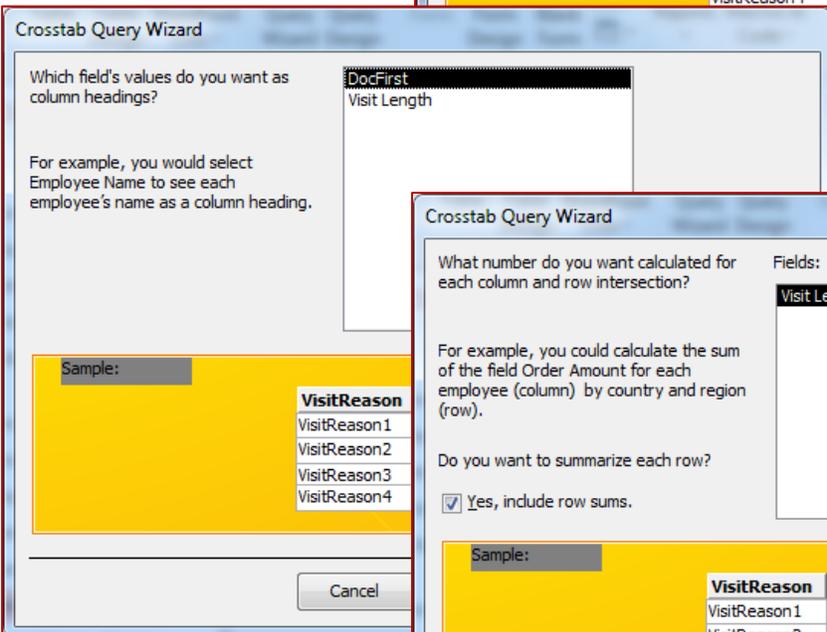
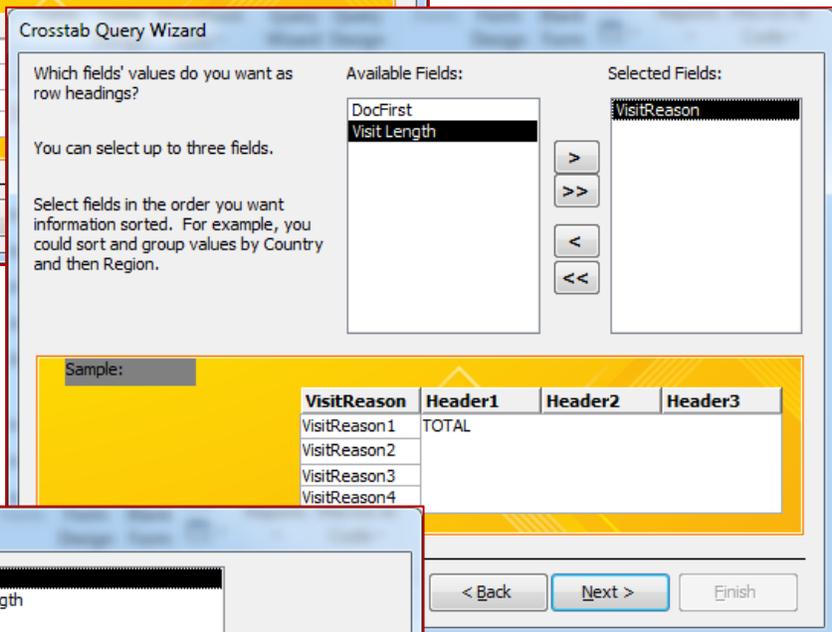


Set the View to Queries or Both and from the list choose

**Query: Doctors by Reason**

Choose the **VisitReason** as the Row Heading.

Choose **DocFirst** as the Column Heading.



Use the Sum Function for the **Visit Length** field

8. Turn on the **Totals** from the **Home** Tab in the Datasheet view to calculate column Totals.

- Under each column change the Total to **Sum**

9. Go to the Design view. Notice the new row in the bottom of the query for **Crosstab**.

- There can only be one **Value** field, and it must have a mathematical summary: Sum, Avg, Count
- There can only be one **Column Heading** field

The screenshot shows the Microsoft Access ribbon with the 'Totals' button highlighted in a red box. Below the ribbon is a table titled 'Doctors by Reason\_Crosstab Sum'.

Reason	Total Of Visi	Sallye	Samuel	Sidney
Allergies	825	240	285	300
Backache	885	345	330	210
Cold/Flu	960	405	240	315
Followup	795	240	240	315
Headache	45		45	
Heartburn	360	120	180	60
Nausea	915	390	285	240
Physical	915	240	345	330
Sore Throat	1050	270	255	525
<b>Total</b>	<b>6750</b>	<b>2250</b>	<b>2205</b>	<b>2295</b>

The screenshot shows the Design view of a query named 'Doctors by Reason\_Crosstab Sum'. A field list box is open, showing the following fields: DocFirst, VisitReason, and Visit Length. Below the field list is a table defining the query structure.

Field:	VisitReason	DocFirst	Visit Length	Total Of Visit Length: Visit Length
Table:	Doctors by Reason	Doctors by Reason	Doctors by Reason	Doctors by Reason
Total:	Group By	Group By	Sum	Sum
Crosstab:	Row Heading	Column Heading	Value	Row Heading
Sort:				
Criteria:				
or:				

Practice: Make another Crosstab Query

- Based on Query: Doctors by Reason
- Row Heading: VisitReason
- Column Heading: DocFirst
- Value Summary: Avg
- Name the Query: **Doctors by Reason Crosstab Avg**
- Once the query is created, change the format property of the Visit Length fields to Standard
  - There are two of them. The Visit Length and the Total of Visit Length