

Introduction

Access offers several options that let you design and run queries that return exactly the information you're looking for. For instance, what if you need to find out **how many** of something exists within your database? Or what if you would like your query results to automatically be sorted a certain way? If you know how to use Access's query options, you can design almost any query you want.

In this lesson, you'll learn how to **modify** and **sort** your queries within Query Design view, as well as how to use the **Totals** function to create a query that can **perform calculations** with your data. You'll also learn about additional query-building options offered in Access.

Throughout this tutorial, we will be using a sample database. If you would like to follow along, you'll need to download our “More Query Design Options.accdb”. You will need to have Access 2013 installed on your computer in order to open the example.

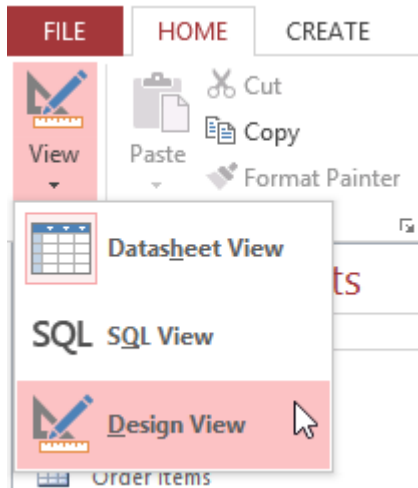
Modifying queries

Access offers several options for making your queries work better for you. In addition to **modifying** your query criteria and joins after you build your queries, you can choose to **sort** and **hide** fields in your query results.

To modify your query:

When you open an existing query in Access, it is displayed in **Datasheet view**, meaning you will see your query results in a table. To modify your query, you must enter **Design view**, the view you used when creating it. There are two ways to switch to Design view:

- On the **Home** tab of the Ribbon, click the **View** command. Select **Design View** from the drop-down menu that appears.



- In the bottom-right corner of your Access window, locate the small **view icons**. Click the **Design View** icon, which is the icon farthest to the right.



Once in **Design view**, make the desired changes, then select the **Run** command to view your updated results.

You may notice that Access also offers **SQL view**. You can ignore this. SQL view allows you to create advanced functions that you will not need to use for this tutorial or for most Access functions.

Sorting queries

Access allows you to apply multiple sorts at once while you're designing your query. This allows you to view your data exactly the way you want.

A sort that includes more than one sorted field is called a **multilevel sort**. A multilevel sort allows you to apply an initial sort, then further organize data with

additional sorts. For instance, if you had a table full of customers and their addresses, you might choose to first sort the records by city, then alphabetically by last name.

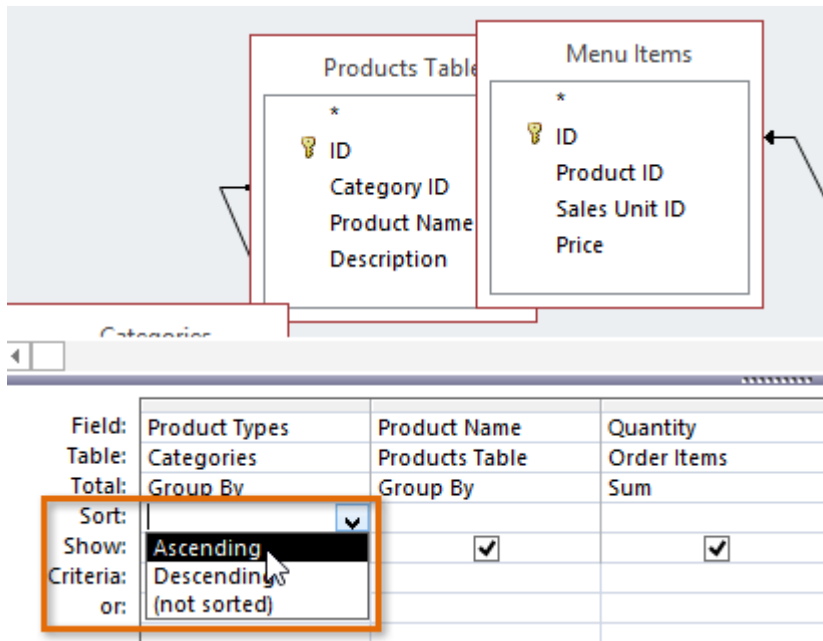
When more than one sort is included in a query, Access reads the sorts from **left to right**. This means the leftmost sort will be applied first. In the example below, customers will be sorted first by the **City** where they live and then by the **Zip Code** within that city.

Field:	City	State	Zip Code
Table:	Customers	Customers	Customers
Sort:	Ascending		Ascending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

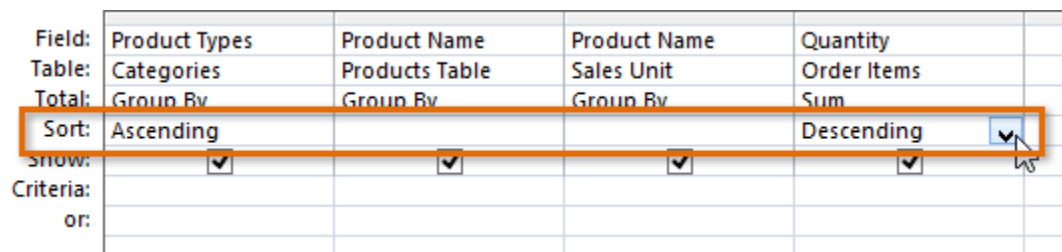
This field is sorted first	This field is sorted second
----------------------------	-----------------------------

To apply a multilevel sort:

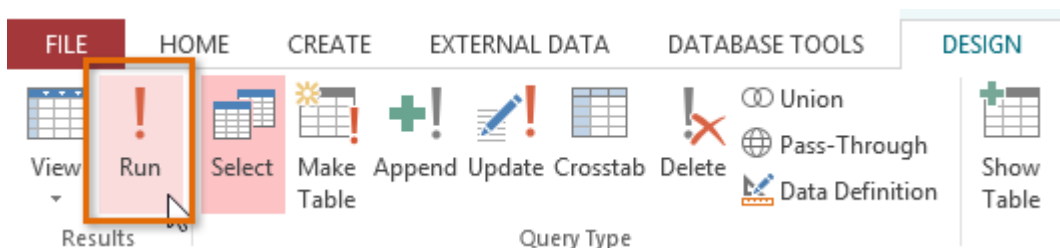
1. Open the query and switch to **Design view**.
2. Locate the field you want to sort first. In the **Sort:** row, click the drop-down arrow to select either an **Ascending** or **Descending** sort.



- Repeat the process in the other fields to add additional sorts. Remember, the sorts are applied from left to right, so any additional sorts must be applied to fields located **to the right** of your primary sort. If necessary, you can **rearrange** the fields by clicking the top of a field and dragging it to a new location.



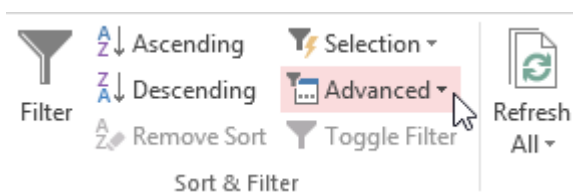
- To apply the sort, click the **Run** command.



- Your query results will appear with the desired sort.

Employee Database Navigation		Ordered Menu Items	
Product Types	Product Name	SumOfQuantity	
Cakes	Cheesecake	20	
Cakes	Buche de Noel (Christmas Cake)- Winter	12	
Cakes	Carrot Cake	9	
Cakes	Black Forest	8	
Cakes	Black Walnut	5	
Cakes	Italian Rum	4	
Cakes	Gingerbread - Winter	4	
Cakes	Coconut	2	
Cakes	French french vanilla	2	
Cakes	German Chocolate	2	
Cakes	Red Velvet	1	
Cakes	Cookies n' Cream	1	
Cookies	Fudge Brownie	7	
Cookies	Fudge Chocolate	6	
Cookies	Ginger Shortbread	6	
Cookies	Chocolate Chip	5	

You can also apply multilevel sorts to tables that don't have queries applied to them. On the **Home** tab on the Ribbon, select the **Advanced** drop-down command in the **Sort & Filter** group. From the menu that appears, select **Advanced Filter/Sort** and create the multilevel sort as you normally would. When you're finished, click the **Toggle Filter** command to apply your sort.



Hiding fields within queries

Sometimes you might have fields that contain important criteria, but you might not need to actually see the information from that field in the final results. For example, take one of the queries we built in our last lesson: a query to find the names and contact information of customers who had placed orders. We

included Order ID numbers in our query because we wanted to make sure we only pulled customers who had placed orders.

However, we really didn't need to see this information in our final query results. In fact, if we were just looking for customer names and addresses, seeing the order number mixed in there might have been distracting. Fortunately, Access makes it easy to **hide** fields while still including any criteria they contain.

To hide a field within a query:

1. Open the query and switch to **Design view**.
2. Locate the field you want to hide.
3. Click the **checkbox** in the **Show:** row to uncheck it.

Field:	City	State	Zip Code	ID	Phone Number
Table:	Customers	Customers	Customers	Orders Table	Customers
Sort:					
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	Not In ("Raleigh")				Like ("919*")
or:					

4. To see the updated query, select the **Run** command. The field will be hidden.

To **unhide** a hidden field, simply return to Design view and click the checkbox in the field's **Show:** row again.

More types of queries

By this point, you should understand how to create a simple one-table or multi-table query using multiple criteria. Additional queries offer you the ability to perform even more complex actions with your database. One of these is the **totals query**, which lets you perform calculations with your data.

Totals queries

Sometimes setting simple criteria won't give you the results you need, especially when you're working with numerical values. You may want to see your query results grouped or counted in some way. For example, let's say we want to find out **how many of each menu item at our bakery has been ordered**—how many Almond Croissants, Apple Pies, and so on. To do this, we could create a totals query to find the **sum** of the **quantities** for each item.

First, the totals query will group all similar menu items from separate orders (for example, Almond Croissants). Then, the Sum function will add the values in the Quantity field to calculate the total number sold for that item.



The **Sum** function helped us find the desired information in this example, but in other situations you may need to use a different function to find the answer you need. There are several functions you can choose from:

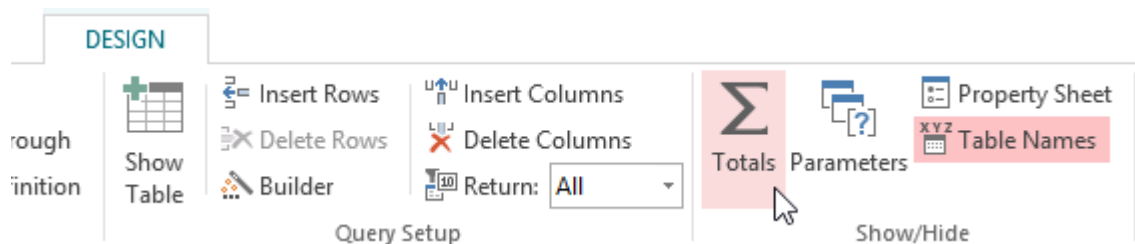
- **Count:** Counts the total number of each item
- **Sum:** Adds the values together
- **Average:** Finds the average of the values
- **Maximum:** Returns the highest value
- **Minimum:** Returns the lowest value
- **First:** Returns the first—or earliest—value
- **Last:** Returns the last—or most recent—value

In our example above, we created a **subtotal** for each menu item in our query. If you wanted to create a **grand total** for all of the items, you would need to add a **totals row**. Review our lesson on [Modifying Tables](#) to learn how.

To create a totals query:

For our example, we want to find the total number we've sold of each of our menu items, so we'll use a query showing us all of the menu items we've sold. If you want to follow along in our database, open the **Menu Items Ordered** query.

1. Create or open a query you want to use as a **totals query**.
2. From the **Design** tab, locate the **Show/Hide** group, then select the **Totals** command.



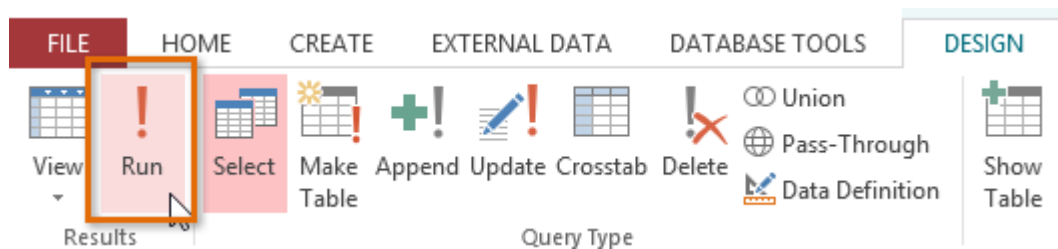
3. A row will be added to the table in the **design grid**, with all values in that row set to **Group By**. Select the cell in the **Total:** row of the field you want to perform a calculation on, then click the drop-down arrow that appears.

Field:	Product Types	Product Name	Product Name	Quantity
Table:	Categories	Products Table	Sales Unit	Order Items
Total:	Group By	Group By	Group By	Group By
Sort:	Ascending	Descending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

4. Select the calculation you want to be performed in that field. In our example, we want to **add** the quantities of products we've sold, so we'll select the **Sum** option.

Field:	Product Types	Product Name	Product Name	Quantity
Table:	Categories	Products Table	Sales Unit	Order Items
Total:	Group By	Group By	Group By	Group By
Sort:	Ascending	Descending		Group By
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sum
Criteria:				Avg
or:				Min
				Max
				Count
				StDev
				Var
				First
				Last
				Expression
				Where

5. When you are satisfied with your query design, select the **Run** command on the **Query Tools Design** tab to **run** the query.



6. The query results will be displayed in the query's **Datasheet view**, which looks like a table. If you want, save your query by clicking the **Save** command on the Quick Access toolbar.

Menu Items Ordered			
Product Types ▾	Products Table.Product Name ▾	Sales Unit.Product Name ▾	SumOfQuantity
Cakes	Cheesecake	Single	
Cakes	Buche de Noel (Christmas Cake)- Winter	Single	
Cakes	Carrot Cake	Single	
Cakes	Black Forest	Single	
Cakes	Black Walnut	Single	
Cakes	Italian Rum	Single	
Cakes	Gingerbread - Winter	Single	
Cakes	Coconut	Single	
Cakes	French french vanilla	Single	
Cakes	German Chocolate	Single	
Cakes	Red Velvet	Single	
Cakes	Cookies n' Cream	Single	
Cookies	Fudge Brownie	One Dozen	
Cookies	Fudge Chocolate	Single	
Cookies	Ginger Shortbread	One Dozen	
Cookies	Chocolate Chip	Single	

More query options

We offer mini-lessons on creating additional types of queries in the last lesson in this tutorial. Below is a list of the queries we currently cover.

- **Parameter query**

A **parameter query** allows you to create a query that can be updated easily to reflect a new criterion, or **search term**. When you open a parameter query, Access will prompt you for a search term and then show you query results that reflect that search.

- **Find duplicates query**

A **find duplicates query** lets you find all **duplicate records** in your database so you can **delete** them. Duplicate records can negatively affect the **integrity** of your database.

Other query-building resources

- Review our [Query Criteria Quick Reference Guide](#) for a list of criteria you can use in building queries. You can also download a **printable version** of the guide.

Challenge!

1. Open an **existing Access database**. If you want, you can use our “More Query Design Options.accdb”.
2. Open the **Customers Who've Ordered from Nearby Towns** query, and switch to **Design view**.
3. Add a **Totals** row to the query.
4. Set the Totals row in the **Orders Table ID** field to **Count**. This will let us count how many orders each customer has placed.
5. In the **Customers** table in the **Object Relationship pane**, double-click the word **City** to add another City field to the design grid below.
6. **Click** and **drag** the City field you just added so it is to the **left** of the **First Name** field. It should now be the **leftmost field** in the design grid.
7. Apply the following multilevel sort:
 - In the leftmost **City** field, apply an **ascending** sort.
 - In the **Last Name** field, apply an **ascending** sort.
8. **Hide** the leftmost **City** field.
9. **Run** the query. If you did it correctly, there should be 14 records in the query results. The first record should look like [this](#).