Using Parameter Queries

[Revised and Updated 21 August 2018]

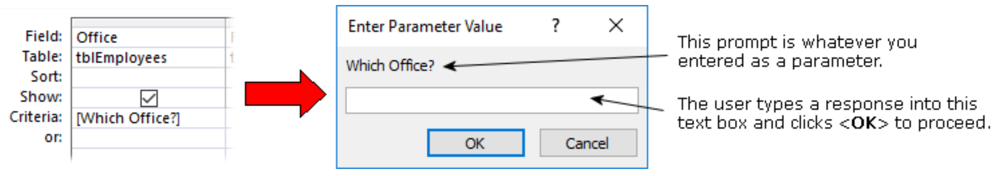
A useful feature of the query is that it can be saved and used again and again, whenever we want to ask the same question. The result we see (the *recordset*) always reflects the most up-to-date information in the database because what you save is the question, not the answer. You just ask the question again by running the query.

Sometimes we want to ask a question time and time again, but the details (the query's *criteria*) may vary. It would be handy to have a way to run a query and make changes to its criteria without having to design a completely new one each time. Access has a tool to solve that problem, the *Parameter Query*. In fact, the parameter query can be any sort of query. You just employ the methods described here to specify the criteria.

When you run a parameter query Access presents you with a dialog box prompting you for the parameter value, which it enters into the appropriate criteria cell. You can have as many parameters as you like in a single query. Here's how it's done.

Entering a Parameter

Instead of typing a value or expression into the criteria cell, type some text enclosed in square brackets (**[ ]**). The text you type will appear as a prompt on a dialog box, so you might want it to be in the form of a question to the user. In this example (*Fig. 1*) the user will be prompted to type the name of the Office when they run the query. The text that the user types will be used as the criteria for that particular field. The dialog box looks like this…



Field:	Office
Table:	tblEmployees
Sort:	
Show:	✓
Criteria:	[Which Office?]
or:	

Enter Parameter Value
?
Which Office?
OK
Cancel
This prompt is whatever you entered as a parameter.
The user types a response into this text box and clicks <OK> to proceed.

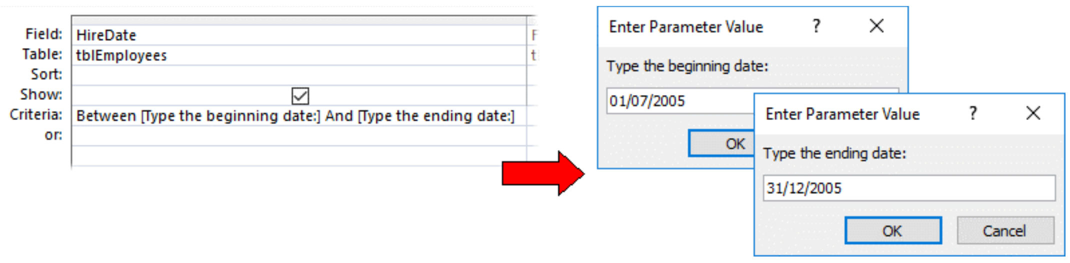
Fig. 1 When you enter a parameter into a query it asks for the criteria when the query is run.

If the user were to type *London* then this query would display all the records with the entry *London* in the *Office* field. The query simply substitutes the text the user enters into the parameter dialog for the parameter you entered into that field's criteria row. Parameters can be used for text, numbers or dates and can be used alone or combined with normally entered criteria.

Using Multiple Parameters

You can enter a parameter almost anywhere you would place a piece of text, number or date in a regular criteria expression. For example, supposing you wanted the query to prompt the user for two dates to define a date range. Instead of typing the actual beginning and end dates into the criteria cell, type a prompt in square brackets. The user will see two separate dialog boxes, each asking for a date (*Fig. 2*). After receiving the dates the query proceeds, inserting the dates into the appropriate places in the criteria expression.

In this example the query would display all the records which contained dates in the range *1 July 2005* to *31 December 2005* in the *HireDate* field...



Field:	HireDate
Table:	tblEmployees
Sort:	
Show:	✓
Criteria:	Between [Type the beginning date:] And [Type the ending date:]
or:	

Type the beginning date: 01/07/2005

Type the ending date: 31/12/2005

Fig. 2 Access displays a separate dialog box for each parameter used.

You can use as many parameters as you want, in as many fields as necessary. The dialog boxes appear in the same order as they do on the QBE grid.

Combining Parameters with Wildcards

A useful feature of the query is its ability to accept wildcards (i.e. an asterisk "*" representing any string of characters; one or more question marks "?", each representing a single character). Wildcards allow you a degree of flexibility when specifying criteria. When you don't know exactly what you are looking for you can use wildcards to give the query a "clue". This method can also be applied to parameter queries, but you need to do a bit more than just add an asterisk or question mark. The correct syntax is as follows…

For a single wildcard:

Like [type prompt here] & "*"

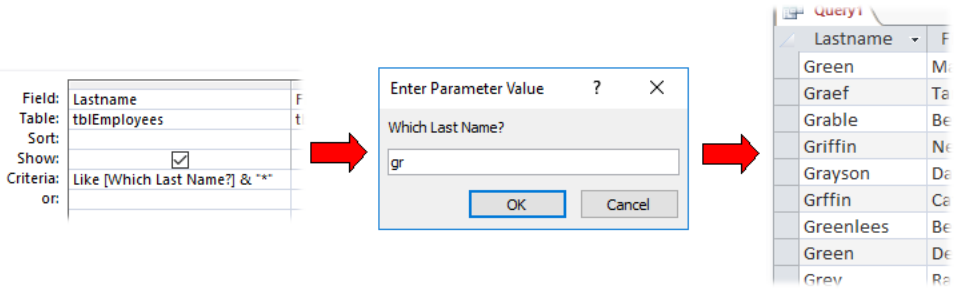
For two wildcards:

Like "*" & [type prompt here] & "*"

When using a single wildcard it can be placed before or after the prompt. You can use asterisks or question marks, or a combination of both.

Using a Single Wildcard

In this example a single wildcard has been used, an asterisk (*Fig. 3*).



Field:	Lastname
Table:	tblEmployees
Sort:	
Show:	☑
Criteria:	Like [Which Last Name?] & "*"
or:	

Enter Parameter Value

Which Last Name?

gr

Lastname	F
Green	M
Graef	Ta
Grable	Be
Griffin	Ne
Grayson	Da
Grffin	Ca
Greenlees	Be
Green	De
Grey	Ra

Fig. 3 A parameter employing a single wildcard.

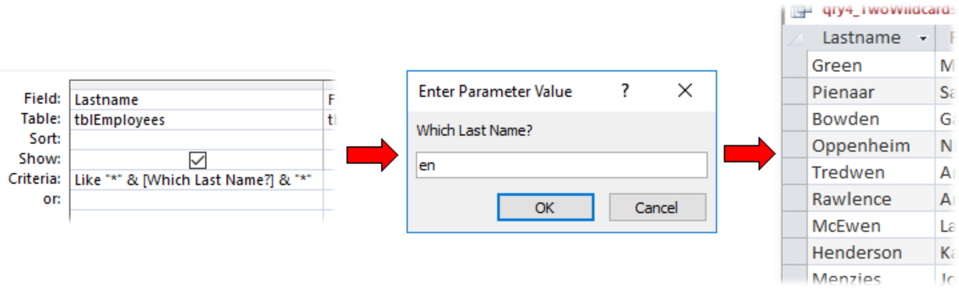
The parameter...

Like [Which Last Name] & "*"

... creates a prompt in which the user can enter the first letter or string of letters of the names they want to see. The user has entered the text "*gr*" causing the query to select records with entries in the *LastName* field of any length starting with the letters "gr".

Using Two Wildcards

In this example (*Fig. 4*) two wildcards have been used, both asterisks.



Field:	Lastname
Table:	tblEmployees
Sort:	
Show:	☑
Criteria:	Like "*" & [Which Last Name?] & "*"
or:	

Enter Parameter Value

Which Last Name?

en

Lastname	F
Green	M
Pienaar	Sa
Bowden	G
Oppenheim	N
Tredwen	A
Rawlence	A
McEwen	La
Henderson	K
Menzies	Jo

Fig. 4 A parameter employing two wildcards.

Fig. 4 A parameter employing two wildcards.

The parameter...

Like "*" & [Which Last Name] & "*"

...creates a prompt in which the user can enter a letter or string of letter that should occur anywhere in the names they want to see. The user has entered the text "*en*", causing the query to select records with entries in the *LastName* field of any length containing with the letters "en" together.

Being Creative with Parameters

You can enter a parameter almost anywhere you would normally enter a specific piece of data in your query criteria. Sometimes the syntax (how you write it out) can be a bit tricky, but persevere until you get the result you need.

Here are a few examples...

Finding records for a specific year (or month) from a collection of dates

Supposing you have a field called *BirthDate* containing a range of dates covering several years. Use the following criteria...

Year([BirthDate])=[Enter a year]

...will create a prompt in which the user can type a year number (e.g. *2005*) to see all the records for people born in that year. I you would rather see records for specific months use...

Month([BirthDate])=[Enter a month from 1-12])

Note that the prompt tells the user to enter a number for the month. Access doesn't understand month names.

Finding records for a specific month and year from a collection of dates

If you want to be more specific and call for a particular month *and* year, the criteria...

Month([BirthDate])=[Enter a month from 1-12] And Year([BirthDate])=[Enter a year]

...will present the user with two dialog boxes, the first asking for a month and the second asking for a year.

Creating a list of records with dates in the last so many days

You may want to view all the invoices generated in a recent period, such as the last 30 days. The criteria...

>Date()-[The last how many days?]

...means "*today minus how many days*". The user enters a number (e.g. *30*) to see a list of dates since that many days before today. The *Date()* part creates the current date so this query is always up-to-date.

What about variable calculations?

You can even include parameters as part of the definition of a new calculated field. (If you want to learn about calculating in Access check out the tutorial *Calculating in Access Queries* at www.fontstuff.com/accessvacctut02.htm)

For example, you have a list of invoices in which there is a field called *TotalGoods* and you need to calculate the discount (or tax or whatever!), but this changes from time to time. You need to create a new calculated field to work out the new figures. Instead of...

Discount: [TotalGoods]*25/100

...which would always calculate a discount of 25% (*note: unlike Excel, Access doesn't understand the % sign as a mathematical operator*). You could substitute the fixed figure with a parameter...

Discount: [TotalGoods]*[What discount rate - percent]/100

...which would prompt the user to enter a figure representing the required discount.

> NOTE: When you run a query that uses a combination of parameters and calculations or functions for the first time, Access will often re-arrange what you have written in the QBE (Query By Example) grid of the query design window. This is so that it can compose the query's SQL correctly. The QBE grid is merely a graphic representation of the SQL language that actually powers the query (check it out in the query's SQL view). You can, of course, compose your criteria in this fashion yourself if you wish. The examples I have given here are simply an easier way for you to enter the parameter criteria.

Asking the Questions in the Right Order

When you create a query using more than one parameter, the user sees the prompts in the order that the fields are arranged in the design view of the query, reading from left to right. You normally arrange the fields in the way in which you want to see the results displayed. But what if you want the prompts to appear in a different order? Get to know the **Query Parameters Window**.

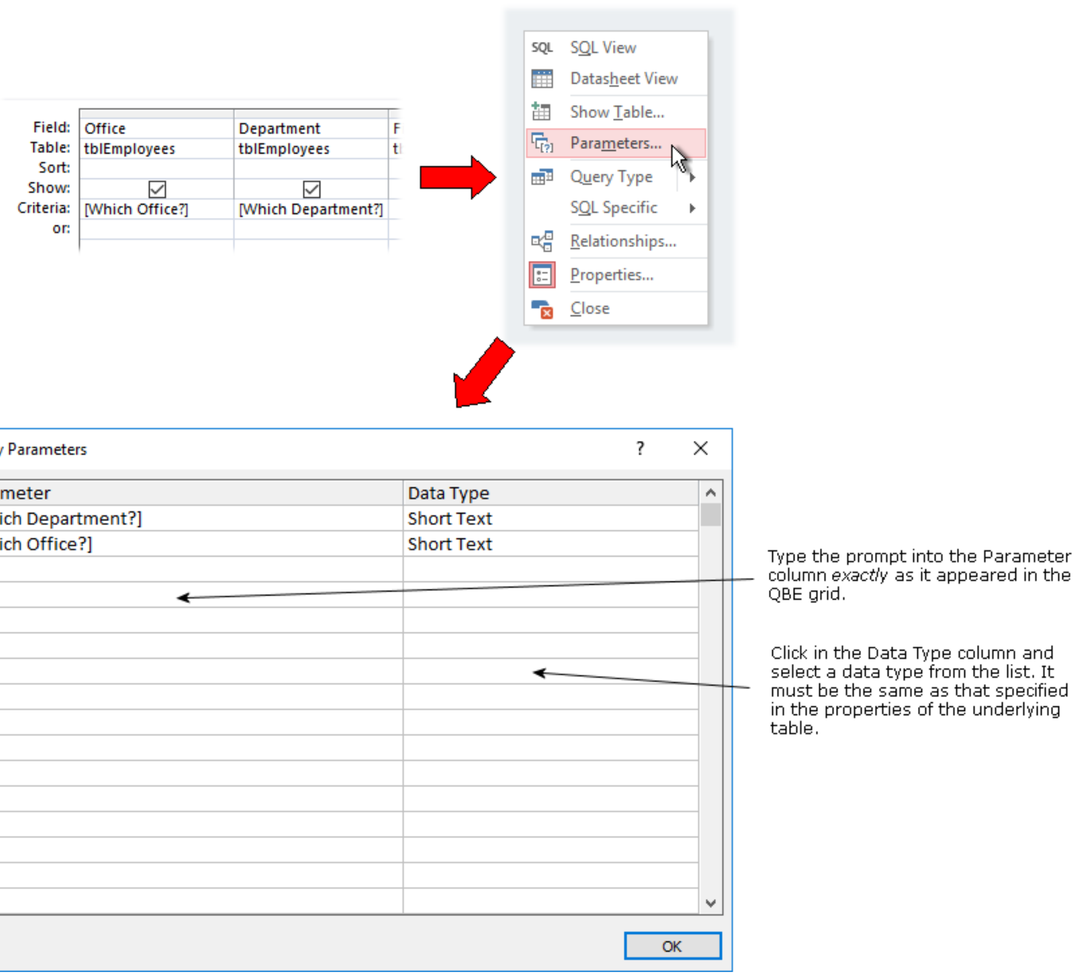
Using the Query Parameters Window

To control the order in which the prompts appear when running a parameter query containing more than one parameter, you can specify the desired order in the **Query Parameters** window. Here's how...

1. In the query design view choose right-click on the upper part of the query design screen and choose **Parameters**... from the context menu to open the **Query Parameters** window.
2. In the **Parameter** column, type the prompt for each parameter *exactly as it was typed in the QBE grid*.
3. In the **Data Type** column specify the kind of data (as defined in the table properties). Pick a type from the list. The default type is *Short Text*.
4. List the parameters in the order in which you want the dialog boxes to appear when the user runs the query.

Click **OK** to accept your entries and close the window.

Here is an example of a query containing two parameters (*Fig. 5*).



Field:	Office	Department
Table:	tblEmployees	tblEmployees
Sort:		
Show:	☑	☑
Criteria:	[Which Office?]	[Which Department?]
or:		

Parameter	Data Type
[Which Department?]	Short Text
[Which Office?]	Short Text

Type the prompt into the Parameter column *exactly* as it appeared in the QBE grid.

Click in the Data Type column and select a data type from the list. It must be the same as that specified in the properties of the underlying table.

Fig. 5 Defining the order in which the parameter prompts appear.

If you didn't specify otherwise, the prompts would appear in the order that the parameters are arranged in the QBE grid reading from left to right. The user would be asked for an *Office* first and then a *Department*. If, however, you make use of the *Query Parameters* window you can choose the order of the prompts. In this example (*Fig. 5*) the parameters have been arranged in a different order so that the user is asked for a *Department* first and then an *Office*.

There is no need to make entries in the *Query Parameters* window if you are happy with the way the query runs, *unless* you are creating a *Crosstab Query* containing parameters, in which case you **must** enter the details of the parameters to make the query run correctly.

What if the User Doesn't Enter Anything?

There is always the possibility that the user will dismiss the parameter dialog without making an entry, either because they don't know what they can type or because they want to see everything. Take a look at my tutorial *Parameter Queries - Handling Null Responses* at www.fontstuff.com/access/acctut07.htm to find out how to prepare for this.

What if the User Doesn't Know What to Type?

It's possible that your users won't know all the entries they can make. I'm often asked how to make things easier for them by offering a list from which they can simply choose an item. Take a look at my tutorial on *Customizing Access Parameter Queries* at www.fontstuff.com/access/acctut08.htm to find out how to do this. It shows you how to build your own custom parameter dialog boxes powered by VBA code. You don't need any prior knowledge of programming. The tutorial explains everything step-by-step and it's a great introduction to programming your databases with VBA.

Dos and Don'ts for Parameter Queries

Prompts must not match field names

When you are designing a parameter query, make sure that the prompt is not exactly the same as one of the field names. You might be tempted to enter the parameter **[LastName]** to prompt the user to enter a name into the dialog box for the *LastName* field. This won't do! Access uses field names in square brackets for calculations in queries so your entry will not be recognised as a parameter and will not appear as a prompt. If you really want to use the name of the field as a prompt, a simple solution is to turn it into a question by adding a question mark e.g. **[LastName?]**.

Don't use illegal characters

You can type just about anything for the prompt, but you mustn't use the period (**.**) exclamation mark (**!**) square brackets (**[]**) or the ampersand (**&**). Everything else is OK.

Don't write too much!

You are only allowed one line of text in the prompt dialog box, which amounts to about 40 to 50 characters depending on what you say. Anything extra just gets cut off. Check your work!