

# Microsoft Access Tips and Tricks

## Parameter Queries

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 *dynaset*) 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 from scratch. 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 design 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 a town 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 (Fig. 2)...

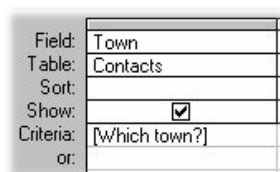


Fig. 1 Entering a parameter

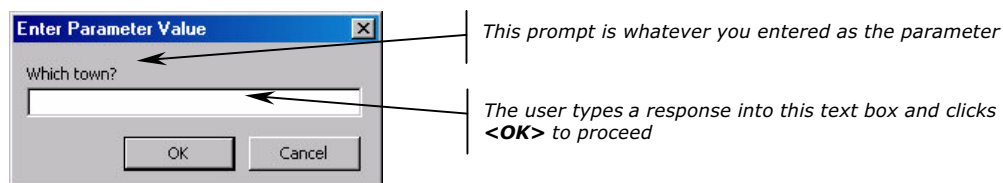


Fig. 2 Parameter dialog box

If the user were to type *London* then this query would display all the records with the entry *London* in the *Town* field.

### Using Multiple Parameters

You can enter a parameter almost anywhere you would place a piece of text, number or date in a regular criterion. For example, supposing you wanted the query to prompt the user for two dates to define a date range.

Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	Between [Type the beginning date:] And [Type the ending date:]
or:	

Fig. 3 Entering multiple parameters

Fig. 4 Multiple parameters create a series of dialogs

Field:	LastName
Table:	People
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	Like [Which Last Name?] & "*"
or:	

Fig. 5 Entering a parameter with a wildcard

Fig. 6 A text entry for the wildcard parameter

LastName
Green
Grable
Green
Green
Grigson
Greenberg

Fig. 7 The result of using a single wildcard

Instead of typing the actual beginning and end dates into the criteria cell, type a prompt in square brackets (Fig. 3).

The user will see two separate dialog boxes, each asking for a date (Fig. 4).

After entering dates the query proceeds, inserted the dates into the appropriate places in the criteria expression.

In this example the query would display all the record which contained dates in the range 1 November 1998 – 30 November 1998 in the Invoice Date field.

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.

#### Example 1: Using a single wildcard

In this example (Fig. 5) a single wildcard has been used, an asterisk.

The parameter...

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

...creates a prompt in which the user has entered the text "gr" (Fig 6), causing the query to select records with entries in the LastName field of any length starting with the letters "gr" (Fig. 7).

Field:	LastName
Table:	People
Sort:	
Show:	<input checked="" type="checkbox"/>
Criteria:	Like "*" & [Which Last Name?] & "*"
or:	

Fig. 8 Entering a parameter with multiple wildcards

Fig. 9 A text entry for the multiple wildcard parameter

LastName
Green
Vincent
MacArena
Green
Green
Moreno
Greenberg

Fig. 10 The result of using multiple wildcards

### Example 2: Using multiple wildcards

In this example (Fig.8) two wildcards have been used, both asterisks.

The parameter...

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

...creates a prompt in which the user has entered the text "en" (Fig. 9), causing the query to select records with entries in the *LastName* field of any length containing with the letters "en" together (Fig. 10).

### Get Creative!

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 *InvoiceDate* containing a range of dates covering several years. Use the following criteria...

**Year([InvoiceDate])=[Choose a year]**

...will create a prompt in which the user can type a year number (e.g. 1998) to see all the records for that year. If you would rather see records for specific months use...

**Month([InvoiceDate])=[Choose 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([InvoiceDate])=[Choose a month from 1-12] And Year([InvoiceDate])=[Choose 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*)

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*). 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.

### ***Asking the questions in the right order***

When you create a query using more than one parameter, the prompts appear 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 this may want the prompts to appear in a different order.

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.

1. In the query design view choose **Query >Parameters...** 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 *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.

Field:	LastName	Department
Table:	People	People
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	[Which Last Name?]	[Which Department?]
or:		

Fig. 11 A query with two separate parameters

Here is an example of a query containing two parameters (Fig. 11).

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 a *Name* 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. 12) the parameters have been arranged in a different order so that the user is asked for a *Department* first and then a *Name*...

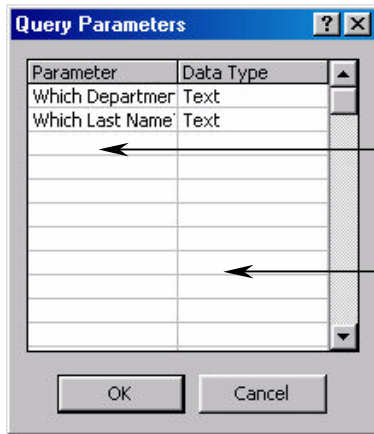


Fig. 12 Entering details into the Query Parameters window

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.

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.

## ***Dos and Don'ts for parameter queries***

### *Prompts mustn't 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! Because Access uses field names in square brackets for calculations in queries, 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... **[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!