

Microsoft Access 2007 – Improving Your Queries

This document explains how you can make simple additions to your Select queries to improve their power and flexibility. It describes how to use Parameters to get criteria from the user, how to use a query to calculate Totals from your data, and how to analyse your data with a Crosstab query.

Parameter Queries

A useful feature of the query is that it can be saved and used again whenever you want to ask the same question. The result you see always reflects the most up-to-date information in the database because the query saves the question, not the answer. You ask the question again by running the query. Sometimes you want to ask a question time and time again, but the criteria may vary. By adding a parameter to a query you can vary its criteria whenever the query is run.

You can add a parameter to any sort of query. 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 to do it...

Entering a Parameter

Instead of typing a value or expression into the criteria cell, type some text enclosed in square brackets ([]) (Fig. 1).

Office	Dep
tblStaff	tblS
<input checked="" type="checkbox"/>	
[Which Office?]	

Fig. 1 Entering the parameter criteria.

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 the user will be prompted to type the name of an office when they run the query (Fig. 2). The text that the user types will be used as the criteria for that particular field. If the user were to type *London* then this query would display all the records with the entry *London* in the *Office* field. The dialog box looks like this...

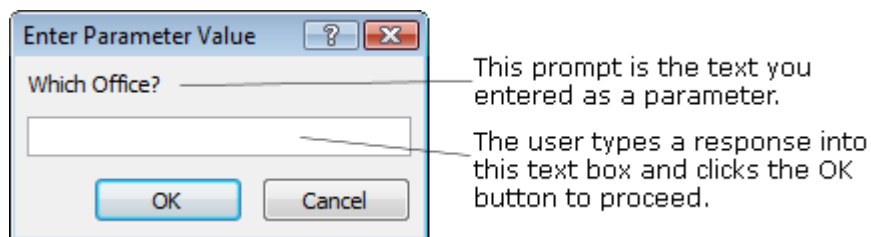


Fig. 2 Access displays a parameter dialog when the query is run.

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...

HireDate	
tblStaff	
Ascending	
<input checked="" type="checkbox"/>	
Between [Hire Date From] And [Hire Date To]	

Fig. 3 Using multiple parameters

Instead of typing the actual beginning and end dates into the criteria cell, type a prompt in square brackets. After entering the first date the user will see a second dialog box (*Fig. 4*). After entering 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 November 2006 - 30 November 2006 in the HireDate field.

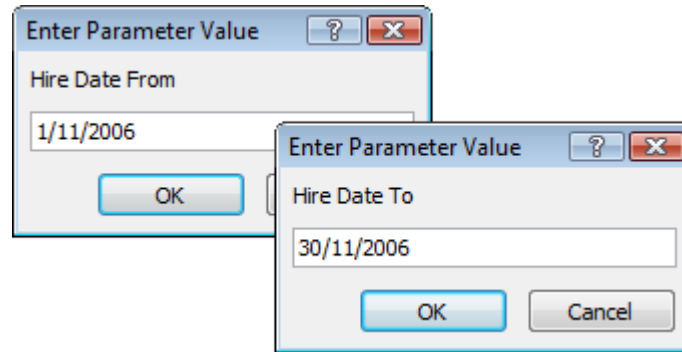


Fig. 4 Access displays a second dialog box.

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

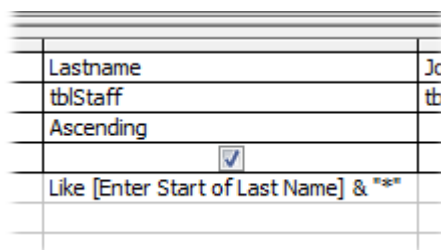


Fig. 5 Using a wildcard with a parameter.

The parameter...

Like [Enter Start of 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 (*Fig. 6*).

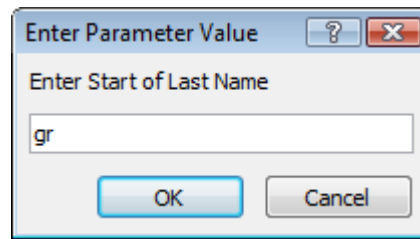


Fig. 6 The user can enter the start letters of a name.

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". Here's the result (Fig. 7) ...

StaffID	FirstName	LastName	JobTitle
48	Betty	Grable	Regional Manager
7	Tamsin	Graef	Head of Administration
673	Tekla	Grainger	IT Engineer
906	Deborah	Gray	Production Assistant
86	David	Grayson	Regional Manager
759	Alison	Green	Sales Executive
202	Deborah	Green	Senior Researcher
366	Laura	Green	Senior Researcher
1	Martin	Green	Chairman
513	Teresa	Green	Senior Researcher

Fig. 7 The query returns names starting with the specified letters.

Showing Totals in Access Queries

New Access users, especially those who are used to keeping their data on an Excel spreadsheet, often ask how they can summarize their records. In Excel you might enter a formula at the foot of a column to sum its values, or maybe use SUMIF or DSUM to examine the data more closely. But an Access table, whilst it may look like a spreadsheet, is just for storing data. When it comes to analyzing your data you need to work with queries.

Adding Totals to a Query

First create a query that includes the fields that you are interested in. In the following example (Fig. 8) I want to create a total value for the purchases made by each company so I have included the *Company* field and created a *Value* field for each item purchased by multiplying the *Quantity* and *Price* fields in my data (NOTE: you can use any field, here it happens to be a calculated field). Running this as a simple select query would show a row for each item purchased by each customer, but the Totals tool allows you to summarise the data by *grouping* the data by one or more fields and calculating *totals* for other fields.

Company	Value
Marsh Ltd	£9,314.19
David Hinds Ltd	£8,266.92
Sara Lee Limited	£7,979.70
Merck Pharmaceuticals Ltd	£7,371.94
Orion Publishing Group	£6,848.58
Cambridge Laboratories Ltd	£6,010.63
Lakeland Ltd	£5,684.41
Diabetes UK Central Office	£5,642.98
Morgan Computers	£5,004.44
Grant Thornton UK LLP	£4,657.02
Personal Computer World	£4,439.81
Hush Beauty Clinic	£4,137.85
Premier Ink Ltd	£3,863.61
British Heart Foundation Shops Division	£3,690.45
Novartis Consumer Health	£3,632.50

Fig. 8 The data has been grouped by Company and a sum of the order values calculated for each one.



To total the data click the **Totals** button on the **Query Design** tab of the Ribbon (Fig. 9). This adds a new row labelled **Total:** to the query design grid. Using this row you must **Group** the data by at least one field (in this case the *Company* field). For the actual totals you have the choice of a number of methods (Sum, Average, Count etc.) so just choose the one you want.

Fig. 9 The Totals button.

You can group a query by several fields and also add several totals if you wish. For example you might want to see the total value of sales along with an average value for each one.

The result is normally sorted automatically, left to right by the grouping field(s) unless you specify otherwise by entering a sort order into the grid yourself.

In the following examples the sales data has been totalled and arranged in descending order. The result is then grouped by *Month* (Fig. 10), this having been extracted from the data in the *OrderDate* field and also by *Salesperson* (Fig. 11).

Month	Value
1	£13,165.58
2	£8,089.25
3	£10,523.74
4	£11,167.76
5	£10,312.33
6	£8,071.87
7	£13,524.15
8	£2,718.90
9	£11,895.24
10	£10,996.28
11	£9,946.49
12	£6,437.28

Fig. 10 The sum of the Value is shown grouped by Month.

Month	Salesperson	Value
1	Rachel Pinkerton	£4,156.80
1	Tina Turner	£1,586.00
1	Pippa Wilde	£1,569.80
1	John Miles	£1,180.57
1	Judith Pentin	£1,007.95
1	Elisabeth Fowler	£952.85
1	Ruth Pham	£738.75
1	Steve Hayim	£578.61
1	Jane Jenkins	£423.55
1	Joan Crawford	£348.65
1	John Hunt	£322.90
1	Fred Quimby	£299.15
2	Fred Quimby	£4,435.45
2	Pippa Wilde	£1,386.55
2	Elisabeth Fowler	£789.20
2	Steve Hayim	£576.85
2	Judith Pentin	£567.40
2	Ruth Pham	£333.80
2	Fred Quimby	£250.70

Fig. 11 The sum of the Values is shown grouped by Month and Salesperson.

The Totals tool can also be used to count records. When doing this it is usually best to count the *Primary Key* field since this is often the only field that can be guaranteed to contain data. Here the *StaffID* field has been counted to provide a summary of staffing levels (Fig. 12).

Field:	Office	Department	StaffID
Table:	tblStaff	tblStaff	tblStaff
Total:	Group By	Group By	Count
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

➔

Office	Department	CountOfStaffID
Amsterdam	Admin	8
Amsterdam	Design	12
Amsterdam	IT	12
Amsterdam	Management	11
Amsterdam	Production	16
Amsterdam	Research	8
Amsterdam	Sales	4
Amsterdam	Shipping	7
Berkeley	Admin	7
Berkeley	Design	6
Berkeley	IT	9
Berkeley	Management	6
Berkeley	Production	25

Fig. 12 The data has been grouped by Office and Department and the number of records counted.

When using the totals tool Access automatically creates a caption for the calculated column such as *SumOfSales* or, as in the above example *CountOfStaffID* (Fig. 12). You can easily change this by modifying the entry in the Field row of the query grid for the calculated column. In the example, changing the entry *StaffID* to *Staffing Level:StaffID* would change the caption of the calculated column to *Staffing Level*.

Crosstab Queries

A Crosstab Query is a special kind of query that summarizes data by plotting one field against one or more other fields. Crosstab Queries can handle large amounts of data with ease and are very easy to create.

Creating a Crosstab Query

A Crosstab Query requires at least three fields to work with, one of which should be either numeric (so that its values can be calculated) or suitable for counting (i.e. to count the number of records).

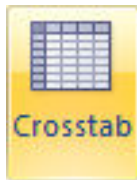


Fig. 13 The Crosstab button.

Enter the fields into the grid and click the **Crosstab** button (Fig. 13) on the **Query Design** tab of the Ribbon. This adds two new rows to the query design grid labelled **Total:** and **Crosstab:**. The Total option works in the same way as when using the Totals tool – you can choose to *Group By* up to four fields but, unlike using the Totals tool on a Select query where you can perform several calculations, a Crosstab query allows you to calculate only one field.

In the Crosstab row you choose **one** field as a *Column Heading* and **up to three** fields as *Row Headings*. When you run the query Access searches each field and groups by the unique values it finds in those fields. For the *Column Heading* field it creates a new column for each value (NOTE: Access can display a maximum of 255 columns. If there are too many unique values in the *Column Heading* field Access will not be able to run the query). Access plots these new columns against the field(s) you have chosen for the *Row Heading(s)* and summarises the calculated data in the middle. In this example (Fig. 14) the Sum of the *Salary* field has been calculated for each *Department* (column heading) for each *Office* (row heading).

Office	Admin	Design	IT
Amsterdam	£219,000	£311,500	£294,500
Berkeley	£198,750	£160,750	£232,750
Birmingham	£222,750	£209,750	£341,000
Brussels	£200,750	£254,000	£211,000
Cardiff	£172,500	£160,750	£248,250
Edinburgh	£220,000	£154,000	£144,250
Indianapolis	£198,750	£177,250	£204,750
London	£323,250	£178,250	£226,000
Manchester	£299,750	£90,500	£166,250
New York	£406,500	£396,250	£225,250
Nice	£251,000	£295,250	£120,000
Paris	£243,750	£232,000	£267,500
Seattle	£199,500	£276,500	£208,250

Fig. 14 The crosstab query has plotted Office against Department and calculated the total Salary for each.

In the next example (Fig. 15) the data has been grouped by three fields, *Office* and *Department* used for row headings and *Gender* for the column headings. The *StaffID* field was used to provide a count of records for the calculated field.

Office	Department	F	M
Amsterdam	Admin	6	2
Amsterdam	Design	8	4
Amsterdam	IT	8	4
Amsterdam	Management	4	7
Amsterdam	Production	9	7
Amsterdam	Research	3	5
Amsterdam	Sales	3	1
Amsterdam	Shipping	1	6
Berkeley	Admin	4	3
Berkeley	Design	2	4
Berkeley	IT	4	5
Berkeley	Management	2	2

Fig. 15 This crosstab query uses three fields to summarise the data.