

Microsoft Access 2007 – Query Criteria Reference

When constructing a query, there is usually a requirement to filter the data somehow so that only certain records are displayed. You need to tell Access what to look for in each field. You do this by defining *criteria* - typing something (called an *expression*) into the Criteria cell of the query or filter grid. If you do not define any criteria for a particular field, Access assumes that you are applying no constraints and will display everything it has. This means that you only have to define criteria for those fields you are interested in.

Criteria are entered in the lower section of the query design window in the QBE grid. The QBE grid is named thus (QBE = *Query By Example*) because at their simplest, criteria take the form of an example of what you want to see in the result. If you want to see results for *London* you enter "London" as your criteria. But query criteria can be much more specific than that, and add a great deal of flexibility to the powerful query tool.

When entering criteria, in addition to typing text, numbers and dates, you need to supply *data qualifiers* that tell Access what sort of data you are entering. There are four main types of data: text, numbers, dates and Boolean (yes/no). Quote marks (" or ') denote text, hash marks (#) denote dates and numbers and Boolean data require nothing. You might also need to enter *operators* of some sort. These might be the usual mathematical operators plus, minus, multiply, divide etc. or conditional operators such as *And*, *Or*, *Between* etc. Add to this a wide range of available functions and the process of specifying query criteria might seem to be getting rather complex! But the rules are quite simple and Access checks your expressions before it allows you to run the query.

If you make an error when entering criteria so that what you type doesn't make sense Access will display an error message. Access will also help by automatically entering data qualifiers and completing criteria expressions for you by adding quote marks when you enter text (*Fig. 1*), hash marks when you enter a date (*Fig. 2*), and even adding appropriate key words to some expressions (*Fig. 3*).

| |
|-------------------------------------|
| FirstName |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| John |



| |
|-------------------------------------|
| FirstName |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| "John" |

Fig. 1 Access adds quote marks to text.

| |
|-------------------------------------|
| HireDate |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| >1/5/06 |



| |
|-------------------------------------|
| HireDate |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| >#01/05/2006# |

Fig. 2 Access adds hash marks to a date.

| |
|-------------------------------------|
| LastName |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| S* |



| |
|-------------------------------------|
| LastName |
| tblStaff |
| |
| <input checked="" type="checkbox"/> |
| Like "S*" |

Fig. 3 Access completes a wildcard expression.

Working With Text

Items of text should be enclosed in quotes (" or ') to distinguish them from other keywords or operators that might be used in the same expression. Access will usually do this for you (*Fig. 1*) but you should always check that it has interpreted your meaning correctly.

Matching Text Items

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | "London" |
| or: | |

"Text"

To match a word or phrase simply type the text you want to match. The query will find all the records that match the text exactly. This example will display all the records that contain the entry *London* in the *City* field.

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | "London" Or "Paris" |
| or: | |

"Text" Or "Text"

To match one of two or more words or phrases, type the text you want to match separated by the word **"Or"**. The query will find all the records that match any of the words or phrases. Enter quote marks yourself if you think the text might confuse the query. This example will display all the records that contain either *London* or *Paris* in the *City* field.

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | "London" |
| or: | "Paris" "Amsterdam" |

"Text""Text""Text"

To match one of several words or phrases, you can type each word or phrase in a new row moving down the column. This gives the same result as using **"Or"** but has the advantage that your criteria might be easier to read. This example will display all the records that contain the entry *London*, *Paris* or *Amsterdam* in the *City* field.

| | |
|-----------|-------------------------------------|
| Field: | Country |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | In ("UK", "USA", "France") |
| or: | |

In ("Text", "Text", "Text")

To match a word or phrase from a list, type the list items separated by commas, and enclose the list in round brackets (parentheses). Access will add the expression **"In"** and place quote marks where needed - you can do this manually if you wish. This example will display all the records that contain *UK* or *USA* or *France* in the *Country* field.

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Not "London" |
| or: | |

Not "Text"

To exclude a word or phrase, use the expression **"Not"** followed by the word or phrase you want to exclude (enclosed in quotes). This example will display records that contain anything other than *London* in the *City* field.

| | |
|-----------|-------------------------------------|
| Field: | Country |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Not In ("UK", "USA", "France") |
| or: | |

Not In ("Text", "Text", "Text")

To exclude a list of words or phrases from the search use the same method as for matching from a list but add the expression **"Not"** at the beginning. This example will display all records that contain anything other than *UK* or *USA* or *France* in the *Country* field.

Using Wildcards

A *wildcard* is a special character that can stand for either a single character or a string of text. Wildcards are useful when you want the query to look for a range of different possible values, and also when you are not certain exactly what you are looking for but can give the query some clues to work with.

The two wildcards we commonly use are the asterisk or star (*) and the question mark (?). The asterisk (*) represents any string of text from nothing up to an entire paragraph or more. The question mark (?) represents a single character only (although you could use, for example, two question marks to represent two unknown characters).

For example:

- **Yor*** would find *York*, *Yorkshire* and *Yorktown* but not *New York*.
- **Mar?** would find *Mark* but not *Mario*, *Martin* or *Omar*.
- **F*d** would find *Fred* and *Ferdinand* but not *Frederick*.

| | |
|-----------|-------------------------------------|
| Field: | Company |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Like "S*" |
| or: | |

Like "Text*"

To match text starting with a particular letter or string type the letter or string of text followed by an asterisk. Access will add the expression "Like" and place quotes around your typing. This example will display all records that have an entry starting with S in the *Company* field.

| | |
|-----------|-------------------------------------|
| Field: | Company |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Like "*Inc" |
| or: | |

Like "*Text"

To match text ending with a particular letter or string type an asterisk followed by a letter or string of text. This example will display all records that have an entry ending with *Inc* in the *Company* field.

| | |
|-----------|-------------------------------------|
| Field: | Company |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Like "[A-D]*" |
| or: | |

Like "[Letter-Letter]*"

To match text starting with letters within a certain range you must type the entire expression as shown (this one is too complicated for Access to work out what you want. This example will display all the records with entries starting with the letters A - D in the *Company* field.

You can often get the same results by using mathematical operators such as greater than (>) and less than (<). These are normally used for specifying numbers and dates but can also be used for text. For example:

- **<"N"** would find all entries beginning with a letter lower than the letter *N* in the alphabet. In other words, all entries starting with the letters A - M.
- **>"F" And <"H"** would find all entries beginning with the letters *F* and *G*.

Working With Numbers

When working with numbers we normally use the mathematical operators to define the range of numbers from which we want to select.

For example, where *n* represents a number:

- **<n** finds values less than *n*
- **>n** finds vales greater than *n*
- **>=n** finds values greater than or equal to *n*
- **<>n** finds vales not equal to *n*

It is important that your field type is correctly defined as a Number field for numerical queries to work properly. Here are some examples...

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | 385 |
| or: | |

n

To match a number simply type the number that you want the query to find. This example will display the record(s) with the entry *385* in the *ContactID* field.

As with text, you can **match one of a list of numbers** by entering the numbers in a column, using the "Or" operator or using the **In()** function as illustrated below...

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | 100 |
| or: | 105 110 |

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | 100 Or 105 Or 110 |
| or: | |

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | In (100,105,110) |
| or: | |

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | <1000 |
| or: | |

<n

To find values less than a certain number type a *less than* sign (<) followed by the number. This example will display all records with an entry less than *1000* in the *ContactID* field. You can also use the *greater than* sign (>) or combine these with an equals sign to specify *greater than or equal to* (>=) or *less than or equal to* (<=).

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Between 500 And 700 |
| or: | |

Between x And y

To find values in a range of numbers you can use the *Between x And y* expression where x and y represent the numbers at opposite ends of the range. This example will display all records with entries falling within the range *500-700* in the *ContactID* field.

| | |
|-----------|-------------------------------------|
| Field: | ContactID |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | >=500 And <=700 |
| or: | |

This method is inclusive of the values at each end of the range and returns the same result as using the mathematical operators *>=x And <=y*.

Working With Dates

Dates behave the same way as numbers, so you can use some of the same techniques when constructing your date query or filter. Remember, for dates to be treated properly by Access it is important that your field type has been correctly defined as a Date/Time field. It doesn't matter how you enter the date, as long as you use a recognised format. The date will be displayed in whatever format you chose when you created the table or you specified for that field of the query.

When you enter a date in the criteria cell you can use any standard date format, but each date must be enclosed by hash marks (#). For example:

- **<#1/1/2005#** finds dates earlier than 1 January 2005
- **=#27-Sep-50#** finds dates equal to 27 September 1950
- **Between #5/7/2005# And #10/7/2005#** finds dates no earlier than 5 July 2005 and no later than 10 July 2005

If you enter a valid date without using hash marks, Access will usually recognise it and add them for you, but sometimes it doesn't (e.g. when the dates are entered as the arguments in a function). It is always advisable to check that Access has interpreted your entry correctly.

When entering dates into the QBE grid use the date convention specified in your computer's Regional Settings for Windows. The examples here are in the European *day/month/year* format. If you live in the United States you would normally use the US *month/day/year* format. Whatever format you use, Access converts the entry to the US format when it constructs the SQL statement behind the query so, if you ever work on your query in *SQL View* remember to use the correct format.

Here are some more examples...

Matching Specific Dates

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | #06/12/2005# | |
| or: | | |

#Date#

To match a specific date enter the date enclosed by hash marks (#). The first example will display all the records with entries for 6 December 2005 in the *InvoiceDate* field.

Where you have a choice of dates you can use the same syntax as for text using the "Or" operator:

| | | |
|-----------|-------------------------------------|---------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | |
| Criteria: | #06/12/2005# Or #13/12/2005# | |
| or: | | |

#Date# Or #Date#

Or enter the dates in a column (remembering the implication this has on criteria for other fields). Alternatively, use the **In()** function to specify a list of dates:

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | #06/12/2005# | |
| or: | #13/12/2005# | |
| | #20/12/2005# | |

In(#Date#, #Date#, #Date#)

| | | |
|-----------|-------------------------------------|---------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | |
| Criteria: | In (#06/12/2005#, #13/12/2005#) | |
| or: | | |

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | Date() | |
| or: | | |

Date()

To match today's date enter the expression **Date()** which means "today". This example will display all the records with entries for the current date in the *InvoiceDate* field.

Specifying a Date Range

| | | |
|-----------|---------------------------------------|---------------|
| Field: | InvoiceDate | InvoiceNumber |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | |
| Criteria: | Between #03/11/2005# And #12/11/2005# | |
| or: | | |

Between #Date# and #Date#

The simplest way of specifying a date range is to use the **Between And** syntax with the first and last dates of the required range. This syntax is inclusive of both dates so both the start and finish dates will be included in the results.

An alternative is to use the mathematical operators to specify *greater than or equal to*

| | |
|-----------|-------------------------------------|
| Field: | InvoiceDate |
| Table: | tblInvoices |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | >=#03/11/2005# And <=#12/11/2005# |
| or: | |

the first date and *less than or equal to* the last date:

>=#Date# And <=#Date#

Both methods return the same result.

Calculated Date Criteria

Access has a wide range of date functions which can be used to construct criteria expressions:

| | | |
|-----------|-------------------------------------|-------------------------------------|
| Field: | InvoiceDate | InvoiceNum |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criteria: | <Date()-30 | |
| or: | | |

<Date()-30

This expression looks for dates more than 30 days old. It uses the **Date()** function to create today's date, subtracts 30 from the result (i.e. 30 days *before* today) and uses the *less than* operator (<) to specify dates prior to (i.e. older than) the resulting date. Since no specific date is "hard-coded" into the expression it will always be current.

| | | |
|-----------|-------------------------------------|-------------|
| Field: | InvoiceDate | InvoiceNu |
| Table: | tblInvoices | tblInvoices |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | |
| Criteria: | Month([InvoiceDate])=3 | |
| or: | | |

Month([FieldName])=MonthNumber

To match a specified month this expression uses the **Month()** function. The example returns records with dates for month 3 (i.e. *March*). Note that the name of the current field is entered into the Month function. As you would expect, the months are numbered 1-12.

| | | |
|-----------|-------------------------------------|-----------|
| Field: | InvoiceDate | InvoiceN |
| Table: | tblInvoices | tblInvoic |
| Sort: | | |
| Show: | <input checked="" type="checkbox"/> | |
| Criteria: | Year([InvoiceDate])=2005 | |
| or: | | |

Year([FieldName])=Year

To match a particular year use the **Year()** function, entering the name of the current field and the required year number. This example will display all the records with a date in *2005* in the *InvoiceDate* field.

| | |
|-----------|-------------------------------------|
| Field: | InvoiceDate |
| Table: | tblInvoices |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Year([InvoiceDate])=Year(Now()) |
| or: | |

Year([FieldName])=Year(Now())

To match the current year use the **Year()** function to require that the year of the record matches the current year. This example will display all the records with entries for the current year in the *InvoiceDate* field.

Working With Yes/No Data

As you would expect, a field with a "Yes/No" data type can only accept the entries *Yes* or *No* but this can be expressed in three different ways: *Yes/No*, *True/False* or *-1/0*. You can use any of these expressions as criteria when querying a Yes/No field.

Excluding Things

Sometimes you want to specifically exclude criteria from your search. This is done with the expression **Not**. This expression can be used on its own or in combination with other expressions. For example:

- **Not "text"** finds all records except those matching the specified text.
- **Not Like "X*"** finds all records except those starting with the specified letter (or string of text).

Here are some more examples:

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Not "London" |
| or: | |

Not "Text"

To exclude specific records from the search use the expression **Not** followed by the text which matches those records you want left out. The text needs to be between quotes as shown here - Access will normally do that for you. This example will find all records for contacts in towns *other than* London.

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Not Like "L*" |
| or: | |

Not Like "Text*"

You can use wildcards with the **Not** expression, which then becomes **Not Like** followed by your wildcard criteria. Here is just one example. This example will find all records for contacts in towns starting with letters *other than* L.

| | |
|-----------|-------------------------------------|
| Field: | City |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Like "L*" And Not "London" |
| or: | |

And Not "Text"

The **Not** expression can be used in combination with other expressions, when it becomes **And Not** followed by the text you want to exclude from your search. This example will find all records for contacts in towns starting with the letter L but will exclude those in London.

Finding Empty Fields

A query can be used to find records where specific fields are empty. To do this you use the expression **Is Null**. Conversely, to find records for which specific fields are not empty you use the expression **Is Not Null**. The expression **Null** simply means "nothing".

If you have made use of the "allow zero length" field property you can search for zero length entries. Sometimes you want to distinguish between, for example, records for which you don't happen to have the particular piece of information for a certain field and those for which you know there definitely isn't any information available. Is the Fax Number field empty because you don't know the person's fax number or is it because they don't have a fax? Either way you can't type a fax number into the field. It has to be left empty. Well, not exactly...

You can make a "zero length entry" (providing this feature has been enabled in the properties of the field - in the table's design view). To do this when entering data type two double-quote marks together without a space between, like this... "". When you leave the field the quote marks disappear and the field looks just like any other empty field - except Access knows it contains a zero length entry. You can search for zero length entries with a query. It is important to remember that if you make use of zero length entries, **Is Null** will *not* find them. It regards them as a piece of text and therefore a field containing a zero length entry is not empty, it just doesn't contain any data. Confused? Read it again then try it out - it does make sense eventually!

Here are some examples:

| | |
|-----------|-------------------------------------|
| Field: | Fax |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Is Null |
| or: | |

Is Null

To find empty fields use the **Is Null** expression. This looks for fields that contain no data. This example will find all records for contacts whose fax number has not been recorded.

| | |
|-----------|-------------------------------------|
| Field: | Fax |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | Is Not Null |
| or: | |

Is Not Null

To find fields that are not empty use the **Is Not Null** expression. This looks for fields that contain data. If there is something in the field the record will be shown. Note that **Is Not Null** will find fields containing zero length entries. (If you want to leave them out try excluding them with the **And Not** expression.) This example finds all records for contacts whose fax number has been recorded.

| | |
|-----------|-------------------------------------|
| Field: | Fax |
| Table: | tblContacts |
| Sort: | |
| Show: | <input checked="" type="checkbox"/> |
| Criteria: | "" |
| or: | |

""

To find zero length entries use "" expression. This looks for zero length entries in the specified field. This example would find, depending on why you had made use of the zero length entry feature, all records for contacts who did not have a fax.

Summary

As I said at the beginning, this is not an exhaustive list of query criteria. Many of these expressions can be combined to create more complex criteria. You can use calculations to construct criteria. The scope is almost limitless. Use your imagination and see what you get! Above all, remember that Access is logical. If you don't get the result you were expecting, read the grid a line at a time (which is what Access does) and see if it makes sense. Sometimes it helps to go and check out the SQL (the language Access uses to write the query - SQL stands for *Structured Query Language*). You can view the SQL by clicking the **SQL View** button on the toolbar.