Access Tips

| Home | Excel | Access | Word | Outlook | FrontPage | VBA | Downloads | Index |

Query Criteria...

- Access Index
- Printer friendly version

Access Query and Filter Criteria

When constructing a query or a filter, you need to tell Access what to look for in each field. You do this by defining criteria - typing something (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.

Here are some examples of the more common types of criteria. Often Access will complete the expression so that you need only type the text you want to match. However, sometimes Access has a choice so you should always check that what Access has written is the same as you intended. If what you type doesn't make sense to Access, you will see an error message.

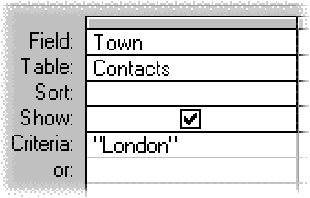
The list of examples below is not exhaustive. Try using combinations of different expressions and see what you get. Also, don't immediately assume that you have made a mistake if you get no records when you run the query or filter. It means that Access can't find anything to match your criteria. That may be because you've asked for something impossible, but it could equally mean that your criteria were perfectly OK but there simply aren't any matching records.

This tutorial is arranged in the following sections:

- Matching Text
- Using Wildcards
- Working with Numbers
- Working with Dates
- Excluding Things
- Finding Empty Fields

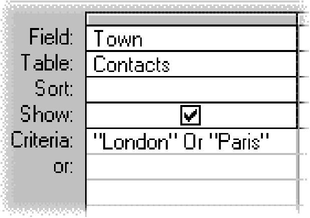
Matching Text

When you enter text into the criteria cell your text should be enclosed in quotes ("") to distinguish it from other expressions and operators that you may need to add.



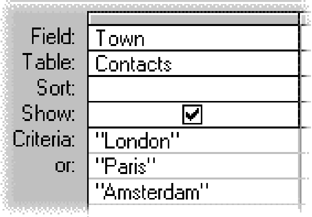
Field:	Town
Table:	Contacts
Sort:	
Show:	☑
Criteria:	"London"
or:	

"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. Access will add the quote marks at each end. It is only necessary to enter the quotes yourself if you type text that might confuse the query. For example you may want to type a phrase that contains the words "and" or "or". Access would normally interpret these words as instructions. You can manually insert the quote marks at each end of the phrase to make sure the criterion means what you intend it to. This example will display all the records that contain the entry *London* in the *Town* field.



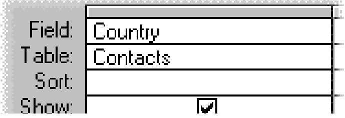
Field:	Town
Table:	Contacts
Sort:	
Show:	☑
Criteria:	"London" Or "Paris"
or:	

"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 *Town* field.



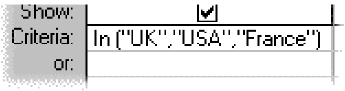
Field:	Town
Table:	Contacts
Sort:	
Show:	☑
Criteria:	"London"
or:	"Paris"
	"Amsterdam"

"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 *Town* field. Note: If this method is combined with criteria for other fields those criteria must be repeated for each row.

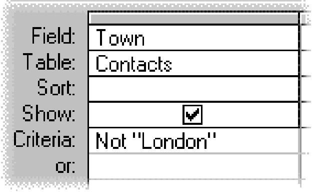


Field:	Country
Table:	Contacts
Sort:	
Show:	☑

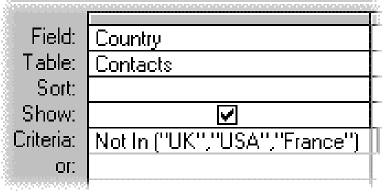
In ("Text", "Text", "Text"...)
To match a word or phrase from a list, type the list items separated by commas, and enclose the list in round


Show: ☑
Criteria: In ("UK","USA","France")
or:

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: Town
Table: Contacts
Sort:
Show: ☑
Criteria: Not "London"
or:

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


Field: Country
Table: Contacts
Sort:
Show: ☑
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.

^ top

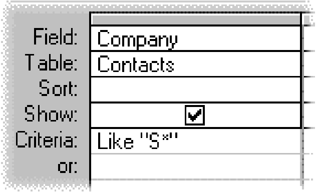
Using Wildcards

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.

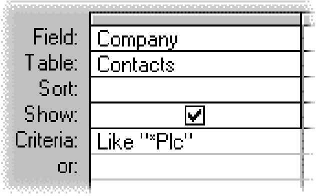
The two wildcards we commonly use are the asterisk or star (*****) and the question mark (**?**).The asterisk (*****) represents any string of text from nothing up to an entire paragraph or more. The question mark (**?**) represents a single character only (although you could use, for example, two question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

For example:

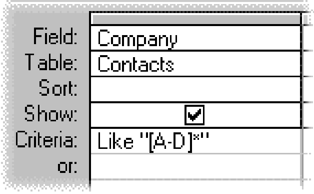
- **Yor*** would find *York*, *Yorkshire* and *Yorktown* but not *New York*.
- **Mar?** would find *Mark* but not *Mario*, *Martin* or *Omar*.
- **F*d** would find *Fred* and *Ferdinand* but not *Frederick*.


Field: Company
Table: Contacts
Sort:
Show: ☑
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: Contacts
Sort:
Show: ☑
Criteria: Like "*Plc"
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 *Plc* in the *Company* field.


Field: Company
Table: Contacts
Sort:
Show: ☑
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*.

^ top

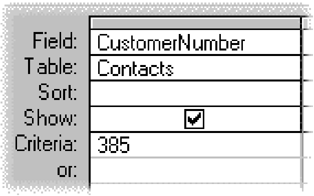
Working with Numbers

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 X represents a number:

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

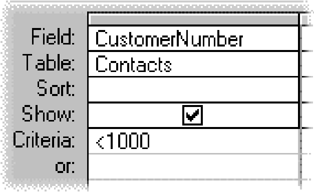
It is important that your field type is correctly defined as a Number field for numerical queries to work properly. Here are some examples...



Field:	CustomerNumber
Table:	Contacts
Sort:	
Show:	☑
Criteria:	385
or:	

X

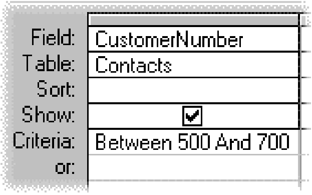
To match a number simply type the number that you want the query to find. This example will display the record(s) with the entry *385* in the *CustomerNumber* field.



Field:	CustomerNumber
Table:	Contacts
Sort:	
Show:	☑
Criteria:	<1000
or:	

<X

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 *CustomerNumber* field.



Field:	CustomerNumber
Table:	Contacts
Sort:	
Show:	☑
Criteria:	Between 500 And 700
or:	

Between X And Y

To find values in a range of numbers type the expression shown 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 *CustomerNumber* field.

^ top

Working with Dates

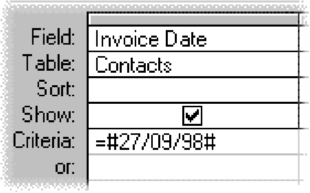
Dates behave the same way as numbers, so you can use some of the same techniques when constructing your date query or filter. Remember, for dates to be treated properly by Access it is important that your field type has been correctly defined as a Date/Time field. It doesn't matter how you enter the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

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:

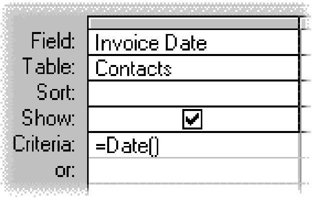
- **<#1/1/98#** finds dates earlier than 1 January 1998
- **=#27-Sep-50#** finds dates equal to 27 September 1950
- **Between #5/7/98# And #10/7/98#** finds dates no earlier than 5 July 1998 and no later than 10 July 1998

Here are some more examples...



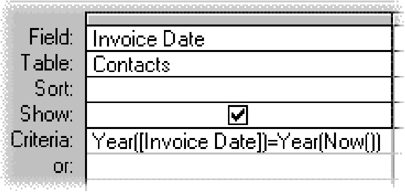
Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	☑
Criteria:	=#27/09/98#
or:	

=#Date#
To match a particular date type the date enclosed by hash marks (#). This example will display all the records with entries for *27 September 1998* in the *Invoice Date* field.



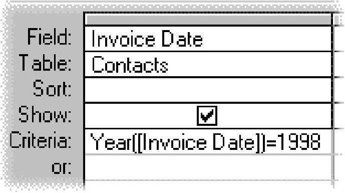
Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	☑
Criteria:	=Date()
or:	

=Date()
To match today's date type the expression shown. **Date()** means "*today*". This example will display all the records with entries for the current date in the *Invoice Date* field.



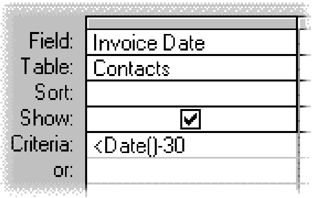
Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	☑
Criteria:	Year([Invoice Date])=Year(Now())
or:	

Year([Fieldname])=Year(Now())
To match the current year type the expression shown, entering the name of the current field in square brackets where indicated. This example will display all the records with entries for the current year in the *Invoice Date* field.



Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	☑
Criteria:	Year([Invoice Date])=1998
or:	

Year([Fieldname])=*Year*
To match a particular year type the expression shown, entering the name of the current field in square brackets where indicated and the required year in place of Year. This example will display all the records with a date in *1998* in the *Invoice Date* field.



Field:	Invoice Date
Table:	Contacts
Sort:	
Show:	☑
Criteria:	<Date()-30
or:	

<Date()-30
To match a particular calculated date range you will need to use a combination of expressions. This expression employs a calculation that subtracts *30* from the current date and also includes the *less than* operator. This example will display all the records with a date more than 30 days old in the *Invoice Date* field.

^ top

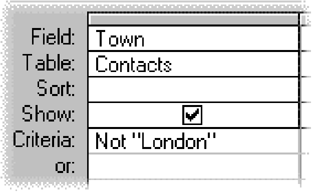
Excluding Things

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:

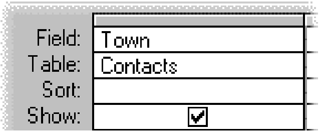
- **Not "text"** finds all records except those matching the specified text.
- **Not Like "X*"** finds all records except those starting with the specified letter (or string of text).

Here are some more examples:



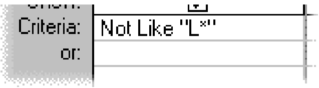
Field:	Town
Table:	Contacts
Sort:	
Show:	☑
Criteria:	Not "London"
or:	

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.



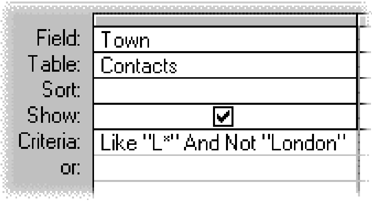
Field:	Town
Table:	Contacts
Sort:	
Show:	☑

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



Criteria:	Not Like "L*"
or:	

contacts in towns starting will letters *other than* L.



Field:	Town
Table:	Contacts
Sort:	
Show:	☑
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.

^ top

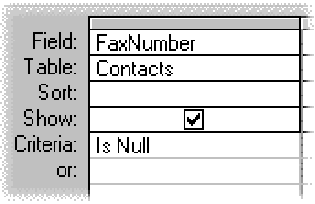
Finding Empty Fields

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...

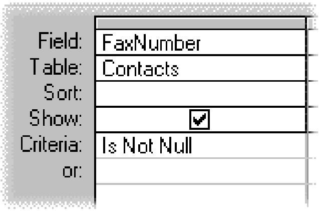
You can make a "zero length entry" (providing this feature has been enabled in the properties of the field - in the table's design view). To do this when entering data type two double-quote marks together without a space between, like this... "". When you leave the field the quote marks disappear and 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:



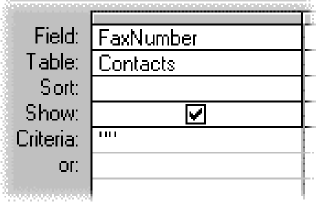
Field:	FaxNumber
Table:	Contacts
Sort:	
Show:	☑
Criteria:	Is Null
or:	

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:	FaxNumber
Table:	Contacts
Sort:	
Show:	☑
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:	FaxNumber
Table:	Contacts
Sort:	
Show:	☑
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.

^ top

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.

^ top

503624