Examples of query criteria

When you want to limit the results of a query based on the values in a field, you use query criteria. A query criterion is an expression that Access compares to query field values to determine whether to include the record that contains each value. For example, = "Chicago" is an expression that Access can compare to values in a text field in a query. If the value for that field in a given record is "Chicago", Access includes the record in the query results.

This topic lists several examples of query criteria. It assumes that you are familiar with designing simple select queries.

To learn more about creating select queries, see the article Select data by using a query.

Introduction to query criteria

A criterion is similar to a formula — it is a string that may consist of field references, operators, and constants. Query criteria are also referred to as expressions in Microsoft Office Access 2007.

The following tables shows some sample criteria and explains how they work.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;25 and &lt;50</td>
<td>This criterion applies to a Number field, such as Price or UnitsInStock. It includes only those records where the Price or UnitsInStock field contains a value greater than 25 and less than 50.</td>
</tr>
<tr>
<td>DateDiff (&quot;yyyy&quot;, [BirthDate], Date()) &gt; 30</td>
<td>This criterion applies to a Date/Time field, such as BirthDate. Only records where the number of years between a person's birthdate and today's date is greater than 30 are included in the query result.</td>
</tr>
<tr>
<td>Is Null</td>
<td>This criterion can be applied to any type of field to show records where the field value is null.</td>
</tr>
</tbody>
</table>

As you can see, criteria can look very different from each other, depending on the data type of the field to which they apply and your specific requirements. Some criteria are simple, and use basic operators and constants. Others are complex, and use functions, special operators, and include field references.

This topic lists several commonly used criteria by data type. If the examples given in this topic do not address your specific needs, you might need to write your own criteria. To do that, you must first familiarize yourself with the full list of functions, operators, special characters, and the syntax for expressions referring to fields and literals. For more information, see the articles listed in the See also section.

Here, you will see where and how you add the criteria. To add a criteria to a query, you must open the query in Design view. You then identify the fields for which you want to specify criteria. If the field is not already in the design grid, you add it by either dragging it from the query design window to the field grid, or by double-clicking the field (Double-clicking the field automatically adds it to the next empty column in the field grid.). Finally, you type the criteria in the Criteria row.

Criteria that you specify for different fields in the Criteria row are combined by using the AND operator. In other words, the criteria specified in the
City and BirthDate fields are interpreted like this:

City = "Chicago" AND BirthDate < DateAdd("yyyy", -40, Date())

1. The City and BirthDate fields include criteria.
2. Only records where the value of the City field is Chicago will satisfy this criterion.
3. Only records of those who are at least 40 years old will satisfy this criterion.
4. Only records that meet both criteria will be included in the result.

What if you want only one of these conditions to be met? In other words, if you have alternate criteria, how do you enter them?

If you have alternate criteria, or two sets of independent criteria where it is sufficient to satisfy one set, you use both the Criteria and the or rows in the design grid.
The City criterion is specified in the Criteria row.

The BirthDate criterion is specified in the or row.

Criteria specified in the Criteria and or rows are combined using the OR operator, as shown below:

City = "Chicago" OR BirthDate < DateAdd("yyyy", -40, Date())

If you need to specify more alternatives, use the rows below the or row.

Before you continue with the examples, note the following:

- If the criteria is temporary or changes often, you can filter the query result instead of frequently modifying the query criteria. A filter is a temporary criterion that changes the query result without altering the design of the query. For more information about filters, see the article Filter: Limit the number of records in a view.

- If the criteria fields don’t change, but the values you are interested in do change frequently, you can create a parameter query. A parameter query prompts the user for field values, and then uses those values to create the query criteria. For more information about parameter queries, see the article Use parameters in queries and reports.

### Criteria for Text, Memo, and Hyperlink fields

If you want to see a demo of criteria that you can use with text fields, see the article Watch this: Specify criteria for a text field.

The following examples are for the CountryRegion field in a query that is based on a table that stores contacts information. The criterion is specified in the Criteria row of the field in the design grid.
A criterion that you specify for a Hyperlink field is, by default, applied to the display text portion of the field value. To specify criteria for the destination Uniform Resource Locator (URL) portion of the value, use the `HyperlinkPart` expression. The syntax for this expression is as follows:

```
HyperlinkPart([Table1].[Field1],1) = "http://www.microsoft.com/"
```

where `Table1` is the name of the table containing the hyperlink field, `Field1` is the hyperlink field, and `http://www.microsoft.com` is the URL you want to match.

<table>
<thead>
<tr>
<th>To include records that...</th>
<th>Use this criterion</th>
<th>Query result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exactly match a value, such as China</td>
<td>&quot;China&quot;</td>
<td>Returns records where the CountryRegion field is set to China.</td>
</tr>
<tr>
<td>Do not match a value, such as Mexico</td>
<td>Not &quot;Mexico&quot;</td>
<td>Returns records where the CountryRegion field is set to a country/region other than Mexico.</td>
</tr>
<tr>
<td>Begin with the specified string, such as U</td>
<td>Like U*</td>
<td>Returns records for all countries/regions whose names start with &quot;U&quot;, such as UK, USA, and so on.</td>
</tr>
<tr>
<td>Do not begin with the specified string, such as U</td>
<td>Not Like U*</td>
<td>Returns records for all countries/regions whose names start with a character other than &quot;U&quot;.</td>
</tr>
<tr>
<td>Contain the specified string, such as Korea</td>
<td>Like &quot;<em>Korea</em>&quot;</td>
<td>Returns records for all countries/regions that contain the string &quot;Korea&quot;.</td>
</tr>
<tr>
<td>Do not contain the specified string, such as Korea</td>
<td>Not Like &quot;<em>Korea</em>&quot;</td>
<td>Returns records for all countries/regions that do not contain the string &quot;Korea&quot;.</td>
</tr>
<tr>
<td>End with the specified string, such as ina</td>
<td>Like &quot;*ina&quot;</td>
<td>Returns records for all countries/regions whose names end in &quot;ina&quot;, such as China and Argentina.</td>
</tr>
<tr>
<td>Do not end with the specified string, such as ina</td>
<td>Not Like &quot;*ina&quot;</td>
<td>Returns records for all countries/regions that do not end in &quot;ina&quot;, such as China and Argentina.</td>
</tr>
<tr>
<td>Contain null (or missing) values</td>
<td>Is Null</td>
<td>Returns records where there is no value in the field.</td>
</tr>
<tr>
<td>Do not contain null</td>
<td>Is Not Null</td>
<td>Returns records where the value is not missing in the field.</td>
</tr>
</tbody>
</table>

**NOTE** When used in an expression, the asterisk (*) represents any string of characters — it is also called a wildcard character. For a list of such characters, see the article [Access wildcard character reference](http://office.microsoft.com/client/helppreview.aspx?AssetID=HA100666111033&...).
values

Contain zero-length strings  

"" (a pair of quotes) Returns records where the field is set to a blank (but not null) value. For example, records of sales made to another department might contain a blank value in the CountryRegion field.

Do not contain zero-length strings  

Not "" Returns records where the CountryRegion field has a nonblank value.

Contains null values or zero-length strings  

"" Or Is Null Returns records where there is either no value in the field, or the field is set to a blank value.

Is not empty or blank  

Is Not Null And Not "" Returns records where the CountryRegion field has a nonblank, non-null value.

Follow a value, such as Mexico, when sorted in alphabetical order  

>= "Mexico" Returns records of all countries/regions, beginning with Mexico and continuing through the end of the alphabet.

Fall within a specific range, such as A through D  

Like "[A-D]" Returns records for countries/regions whose names start with the letters "A" through "D".

Match one of two values, such as USA or UK  

"USA" Or "UK" Returns records for USA and UK.

Contain one of the values in a list of values  

In("France", "China", "Germany", "Japan") Returns records for all countries/regions specified in the list.

Contain certain characters at a specific position in the field value  

Right ((CountryRegion), 1) = 'y' Returns records for all countries/regions where the last letter is "y".

Satisfy length requirements  

Len(CountryRegion) > 10 Returns records for countries/regions whose name is more than 10 characters long.

Match a specific pattern  

Like "Chi??" Returns records for countries/regions, such as China and Chile, whose names are five characters long and the first three characters are "Chi".

NOTE The characters ? and _ when used in an expression, represent a single character — these are also called wildcard characters. The character _ cannot be used in the same expression with the ? character, nor can it be used in an expression with the * wildcard character. You may use the wildcard character _ in an expression that also contains the % wildcard character.

Criteria for Number, Currency, and AutoNumber fields

The following examples are for the UnitPrice field in a query that is based on a table that stores products information. The criterion is specified in the Criteria row of the field in the query design grid.
<table>
<thead>
<tr>
<th>To include records that...</th>
<th>Use this criterion</th>
<th>Query Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exactly match a value, such as 100</td>
<td>100</td>
<td>Returns records where the unit price of the product is $100.</td>
</tr>
<tr>
<td>Do not match a value, such as 1000</td>
<td>Not 1000</td>
<td>Returns records where the unit price of the product is not $1000.</td>
</tr>
<tr>
<td>Contain a value smaller than a value, such as 100</td>
<td>&lt; 100, &lt;= 100</td>
<td>Returns records where the unit price is less than $100 (&lt;100). The second expression (&lt;=100) displays records where the unit price is less than or equal to $100.</td>
</tr>
<tr>
<td>Contain a value larger than a value, such as 99.99</td>
<td>&gt;99.99, &gt;=99.99</td>
<td>Returns records where the unit price is greater than $99.99 (&gt;99.99). The second expression displays records where the unit price is greater than or equal to $99.99.</td>
</tr>
<tr>
<td>Contain one of the two values, such as 20 or 25</td>
<td>20 or 25</td>
<td>Returns records where the unit price is either $20 or $25.</td>
</tr>
<tr>
<td>Contain a value that falls within a range of values</td>
<td>&gt;49.99 and &lt;99.99 -or- Between 50 and 100</td>
<td>Returns records where the unit price is between (but not including) $49.99 and $99.99.</td>
</tr>
<tr>
<td>Contain a value that falls outside a range</td>
<td>&lt;50 or &gt;100</td>
<td>Returns records where the unit price is not between $50 and $100.</td>
</tr>
<tr>
<td>Contain one of many specific values</td>
<td>In(20, 25, 30)</td>
<td>Returns records where the unit price is either $20, $25, or $30.</td>
</tr>
<tr>
<td>Contain a value that ends with the</td>
<td>Like &quot;*4.99&quot;</td>
<td>Returns records where the unit price ends with &quot;*4.99&quot;, such as $4.99, $14.99, $24.99, and so on.</td>
</tr>
</tbody>
</table>
The following examples are for the OrderDate field in a query based on a table that stores Orders information. The criterion is specified in the Criteria row of the field in the query design grid.

**Criteria for Date/Time fields**

To include records that ...

<table>
<thead>
<tr>
<th>Use this criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>To include records that ...</td>
</tr>
<tr>
<td>Exactly match a value, such as 2/2/2006</td>
</tr>
<tr>
<td>Do not match a value, such as 2/2/2006</td>
</tr>
<tr>
<td>Contain values that fall before a certain date, such as 2/2/2006</td>
</tr>
<tr>
<td>Contain values that fall after a certain date, such as 2/2/2006</td>
</tr>
</tbody>
</table>

**NOTE** The characters * and %, when used in an expression, represent any number of characters — these are also called wildcard characters. The character % cannot be used in the same expression with the * character, nor can it be used in an expression with the ? wildcard character. You may use the wildcard character % in an expression that also contains the _ wildcard character.
Contain values that fall within a date range

>2/2/2006# and <2/4/2006#

Returns records where the transactions took place between Feb 2, 2006 and Feb 4, 2006.

You can also use the **Between** operator to filter for a range of values, including the end points. For example, Between #2/2/2006# and #2/4/2006# is the same as >=#2/2/2006# and <=#2/4/2006#.

Contain values that fall outside a range

<2/2/2006# or >2/4/2006#

Returns records where the transactions took place before Feb 2, 2006 or after Feb 4, 2006.

Contain one of two values, such as 2/2/2006 or 2/3/2006

#2/2/2006# or #2/3/2006#

Returns records where the transactions took place on either Feb 2, 2006 or Feb 3, 2006.

Contain one of many values

In (#2/1/2006#, #3/1/2006#, #4/1/2006#)

Returns records where the transactions took place on Feb 1, 2006, March 1, 2006, or April 1, 2006.

Contain a date that falls in a specific month (irrespective of year), such as December

DatePart("m", [SalesDate]) = 12

Returns records where the transactions took place in December of any year.

Contain a date that falls in a specific quarter (irrespective of year), such as the first quarter

DatePart("q", [SalesDate]) = 1

Returns records where the transactions took place in the first quarter of any year.

Contain today's date

Date()

Returns records of transactions that took place on the current day. If today's date is 2/2/2006, you see records where the OrderDate field is set to Feb 2, 2006.

Contain yesterday's date

Date()-1

Returns records of transactions that took place the day before the current day. If today's date is 2/2/2006, you see records for Feb 1, 2006.

Contain tomorrow's date

Date()+1

Returns records of transactions that took place the day after the current day. If today's date is 2/2/2006, you see records for Feb 3, 2006.

Contain dates that fall during the current week

DatePart("ww", [SalesDate]) = DatePart("ww", Date()) and Year ([SalesDate]) = Year(Date())

Returns records of transactions that took place during the current week. A week starts on Sunday and ends on Saturday.

Contain dates that fell during the previous week

Year([SalesDate])* 12 + DatePart("m", [SalesDate]) = Year(Date())* 12 + DatePart("m", Date()) - 1

Returns records of transactions that took place during the last week. A week starts on Sunday and ends on Saturday.

Contain dates that fall during the following week

Year([SalesDate])* 12 + DatePart("m", [SalesDate]) = Year(Date())* 12 + DatePart("m", Date()) + 1

Returns records of transactions that will take place next week. A week starts on Sunday and ends on Saturday.

Contain a date that fell during the last 7 days

Between Date() and Date()-6

Returns records of transactions that took place during the last 7 days. If today's date is 2/2/2006, you see records for the period Jan 24, 2006 through Feb 2, 2006.

Contain a date that belongs to the current month

Year([SalesDate]) = Year(Now()) And Month([SalesDate]) = Month(Now())

Returns records for the current month. If today's date is 2/2/2006, you see records for Feb 2006.

Contain a date that belongs to the previous month

Year([SalesDate])* 12 + DatePart("m", [SalesDate]) = Year(Date())* 12 + DatePart("m", Date()) - 1

Returns records for the previous month. If today's date is 2/2/2006, you see records for Jan 2006.

Contain a date that belongs to the next month

Year([SalesDate])* 12 + DatePart("m", [SalesDate]) = Year(Date())* 12 + DatePart("m", Date()) + 1

Returns records for the next month. If today's date is 2/2/2006, you see records for Mar 2006.

Contain a date that fell during the last 30 or 31 days

Between Date( ) And DateAdd("M", -1, Date( ))

A month's worth of sales records. If today's date is 2/2/2006, you see records for the period Jan 1, 2006 to Feb 2, 2006.

Contain a date that belongs to

Year([SalesDate]) = Year(Now())

Returns records for the current quarter. If today's date is 2/2/2006, you see records for the period Jan 1, 2006 to Feb 28, 2006.
Criteria for other fields

Yes/No fields In the Criteria row, type Yes to include records where the check box is selected. Type No to include records where the check box is not selected.

Attachments In the Criteria row, type Is Null to include records that do not contain any attachments. Type Is Not Null to include records that contain attachments.

Lookup fields There are two types of Lookup fields: those that look up values in an existing data source (by using a foreign key), and those that are based on a list of values specified when the Lookup field is created.

Lookup fields that are based on a list of specified values are of the Text data type, and valid criteria are the same as for other text fields.

The criteria you can use in a Lookup field based on values from an existing datasource depend on the data type of the foreign key, rather than the data type of the data being looked up. For example, you may have a Lookup field that displays Employee Name, but uses a foreign key that is of the Number data type. Because the field stores a number instead of text, you use criteria that work for numbers; that is, > 2.

If you do not know the data type of the foreign key, you can inspect the source table in Design view to determine the data types of the field. To do this:

1. Locate the source table in the Navigation Pane.
2. Open the table in Design view by either:
   - Clicking the table, and then pressing CTRL+ENTER
Right-clicking the table, and then clicking **Design View**.

3. The data type for each field is listed in the **Data Type** column of the table design grid.

**Multivalued fields** Data in a multivalued field are stored as rows in a hidden table that Office Access 2007 creates and populates to represent the field. In query Design view, this is represented in the **Field List** by using an expandable field. To use criteria for a multivalued field, you supply criteria for a single row of the hidden table. To do this:

1. Create a query containing the multivalued field, and open it in Design view.

2. Expand the multivalued field by clicking the plus symbol (+) next to it — if the field is already expanded, this is a minus symbol (-). Just below the name of the field, you will see a field representing a single value of the multivalued field. This field will have the same name as the multivalued field, with the string .Value appended.

3. Drag the multivalued field and its single value field to separate columns in the design grid. If you want to see only the complete multivalue field in your results, clear the **Show** check box for the single value field.

4. Type your criteria in the **Criteria** row for the single value field, using criteria that is appropriate for whatever type of data the values represent.

5. Each value in the multivalued field will be individually evaluated using the criteria you supply. For example, you may have a multivalued field that stores a list of numbers. If you supply the criteria >5 AND <3, any record where there is at least one value greater than 5 and one value less than 3 will match.

**Related Office Online discussions**

Read related questions and answers from other Microsoft Office customers.

- Can you use the Like statement in a crosstab query criteria?
- Troubleshooting parameter prompts that keep popping up???
- To filter or not to filter?
- How can I display data/calculate ranges in a query prompt?

**See Also**

- Access wildcard character reference
- Examples of expressions
- Find records with the most recent or oldest dates
- Find the records with the top or bottom values in a group or field
- Find, hide, or eliminate duplicate data
- Functions (alphabetical)
- Watch this: Specify criteria for a text field

**Was this information helpful?**

[Yes]  [No]  [I don't know]