

## Create Lists Conditions Definitions

**= (Equal to)** - The entire normalized field, including all subfields, must exactly match the characters entered.

**Use for fixed-length fields** -- This relational operator is useful for searching in fixed-length fields. However, it is **NOT** recommended for **searching in variable-length fields**, since any difference between the search statement and the data in the field causes the system to find no match. For example, a search for "AUTHOR = Cartland Barbara" does not retrieve a record containing the MARC 100 field "Cartland, Barbara, |d1902-", since the search statement did not include subfield 'd'. **Use the "has" operator for searching variable-length fields.**

**Use to locate item records that contain a specific course record number** -- When you choose to search on the /6 COURSE ID field, use this relational operator to designate that items retrieved must contain the course record number you specify.

**Use to locate records that do not contain a particular variable-length field** -- You can use this relational **operator to retrieve records that lack a particular variable-length field**. Select this operator and do not specify any characters after the equal sign. For example, the search for "AUTHOR=" retrieves all records that do not contain an author field. **Think of this as a search for "author equals nothing", which retrieves all records with no authors.**

**!= (Not equal to)** -The normalized field does not exactly match the characters entered, that is, it has characters other than or in addition to the specified characters.

**Use for fixed-length fields** -- This relational operator is useful for searching fixed-length fields. For example, a search for "LOCATION != multi" retrieves all records which **do not** have multiple locations. Similarly, the search "LOCATION != main" retrieves records that **do not** have a single location of "main." **However, it also retrieves records that have a location of "multi" with "main" being one of the multiple locations.** To find all records that **do not have "main" as a location**, use the **All fields do not have operator**. You can search for records that have a single location of "main" using the search "LOCATION != multi AND LOCATION = main".

**This operator is NOT recommended for searching variable-length fields.** If you use this operator alone to search variable-length fields, the system can retrieve virtually the entire database. For example, a search for "AUTHOR != Cartland Barbara" retrieves a record containing the MARC 100 field "Cartland, Barbara, |d1902-", because subfield 'd' was not included in the search statement and therefore the two do not match exactly. This search also retrieves every other record in the database that does not have a MARC 100 field that exactly matches "Cartland Barbara." Even when you use it in combination with other, more restrictive, conditions, it is not an effective search tool for variable-length fields, since a single difference between the search statement and the data in the field can cause the system to retrieve an undesired record.

**Use to locate records which do contain a particular variable-length field** -- You can use this relational operator to retrieve records that contain a particular variable-length field. Select this operator and do not specify any characters after "!=". For example, the search for "AUTHOR !=" retrieves all records that do contain an author field. Think of this as a search for "author not equal to nothing", which retrieves all records with an author.

## > (Greater than)

## < (Less than)

These relational operators perform a **character-by-character comparison** between the search statement and the data in the specified field, starting with the left-most character in each. **The system stops the comparison when the first non-matching character is reached.** For the "greater than" operator, the system retrieves a record if the ASCII value of the first non-matching character is greater in the record than in the search statement. For the "less than" operator, the opposite is true. If the search statement and the data from the field are exactly equal, the record is not retrieved. For example, if a copy on order has an estimated price of \$51, a search for "ESTIMATED PRICE < 50" does not retrieve that record, whereas a search for "ESTIMATED PRICE > 50" does retrieve it. Neither search retrieves a copy on order with an estimated price of \$50.

If all characters in a search statement match the data in a record, but the search statement is shorter, the system considers the data in the record to be "greater than" the search statement. Thus, to search for all authors whose name begins with 'a', 'b', or 'c', enter the search statement "AUTHOR < d". Likewise, to find all authors whose name begins with 'm', enter the search "AUTHOR > m AND AUTHOR < n".

## >= (Greater than or equal to)

## <= (Less than or equal to)

These are the same as "greater than" and "less than" described previously, except that exact matches also are retrieved. For example, if a copy on order has an estimated price of \$50, a search for "ESTIMATED PRICE >= 50" retrieves that record, as does a search for "ESTIMATED PRICE <= 50".

## H (Has)

The system retrieves a record if the characters entered are contained, **as a group**, anywhere in any **subfield** of the specified field in the record. **The entire group of characters must match exactly, including spaces, punctuation marks, and subfield delimiters.** The field can contain additional characters. The system examines the entire field, that is, you cannot specify a particular subfield.

**Use for variable-length fields** -- Use the operator to find the entered characters anywhere in any subfield in the specified field. For example, searching for "AUTHOR has land" retrieves the MARC 100 fields "Landman, Paul", "Hollander, John", "Guildhall School (England)", and so forth. **This is the recommended method for searching variable-length fields.** Note that the search for "AUTHOR has Cartland Barbara" does not retrieve a record with the MARC 100 field "Cartland, Barbara", because the comma does not appear in the search statement.

## A (All fields do not have)

This operator **retrieves records that contain the given field**, provided that all occurrences of the given field **do not** have the **specified character or string of characters**.

## O (At least one field does not have)

This operator **retrieves records that contain the given field**, provided that at least **one occurrence** of the given field **does not** have the **specified character or string of characters**.

### **W (Between)**

Use this operator to find a **value between two specified values** (entered values are included in the range). This operator usually is used for **dates or amounts**, for example, "ESTIMATED PRICE between 250 & 500" locates records with estimated prices between \$250 and \$500 (inclusive).

### **N (Not within)**

Use this operator **to find a value that is not between two specified values** (entered values are included in the range). This operator usually is used for **dates or amounts**, for example, "CATDATE not within 01/01/93 & 12/31/93" excludes all items cataloged during 1993.

### **^ (Starts with)**

Use this operator **to find variable-length fields in which a given word or phrase appears** at the beginning of the field. The comparison between the word or phrase you enter and the beginning of the field is case insensitive, thus "Journal of" and "journal of" produce the same results. This operator is not supported for fixed-length fields.

### **\$ (Ends with)**

Use this operator **to find variable-length fields in which a given word or phrase appears at the end of the field**. The comparison is **case insensitive**. Punctuation is included as part of the comparison, so if you enter "cultural studies" and a field ends with "cultural studies." (note the period), then the field would not match. This operator is not supported for fixed-length fields.

### **E (Exist)**

This operator **identifies date fields**, such as CAT DATE or LCHKIN, that have content (date entries).

### **N (Not exist)**

This operator **identifies blank (null) date fields**.

### **T (equals today)**

Retrieves records whose **specified DATE field is equal to the day** on which the search is run.

### **Y (equals yesterday)**

Retrieves records whose **specified DATE field is equal to the day before** the search is run.

### **V (within last week)**

Retrieves records whose **specified DATE field falls within the previous calendar week** (Sunday - Saturday).

### **M (within last month)**

Retrieves records whose **specified DATE field falls within the previous calendar month**.

### **A (is this many days ago)**

Retrieves records whose **specified DATE field is equal to the specified number of days before** the date on which the search is run.

## **B (is this many weeks ago)**

Retrieves records whose **specified DATE field is equal to the specified number of weeks before** the date on which the search is run.

## **C (is this many months ago)**

Retrieves records whose **specified DATE field is equal to the specified number of months before** the date on which the search is run.

## **R (Matches) - Regular Expressions**

The '**r**' **operator enables you to enter a regular expression to search variable-length, and some fixed-length**, fields in database records. Regular expressions are a powerful and complex tool that permit you to specify a pattern of characters for which to search. A regular expression can consist of ASCII characters alone or ASCII characters accompanied by special regular expression symbols.

**Not all fixed-length fields can be searched using regular expressions.** You can search the **LOCATION, COUNTRY, HOME LIBRARY, FUND, VENDOR, LANGUAGE, and BOX NOTE fixed-length fields using regular expressions.** Note that **these fields** are stored as **five characters**, right-padded with spaces, so **"nor" is actually stored as "nor "** (with two trailing spaces). Take care in constructing your regular expression to match this format.

When you search a field, the regular expression you use **must appear somewhere in the field exactly as you specify.** The regular expression does not need to match the entire field, as required by the "equal to" operator.

For example, the search:

TITLE matches data processing  
matches records with TITLE fields containing "Modern data processing," "Data processing for beginners," and "Stretching man's mind :|ba history of data processing."

+ The **plus sign** is a special regular expression symbol that **allows you to match one or more of the preceding characters.**

For example, a search for:

DESCRIPT matches x+i+  
matches records with DESCRIPT fields containing "xi, 318 p. ;|c18 cm.," "xii, 610, [29] p.," "xix, 374 p.," and "xxii, 314 p."

\* The **asterisk** allows you **to match zero or more of the preceding characters.**

For example, a search for:

TITLE matches dea\*f  
matches records with TITLE fields containing "Eating defensively" and "Deafness in infancy and early childhood."

. The **period** is a **special symbol that matches any single character, including letters, numbers, spaces, and punctuation marks.**

For example, a search for:

AUTHOR matches j.n

matches records with AUTHOR fields containing "Jones, Aubrey B," "Heise, Jon O.," and "Janaro, Richard Paul." You can combine the period with the plus sign or asterisk to search for more than a single character.

For example,

AUTHOR matches j.\*n

matches all of the records of the previous search as well as records containing "Couger, J. Daniel," "Diebold, John," and "Rosenberg, Jerry Martin."

[ ] **Square brackets** are **special regular expression symbols that enable you to define patterns** that match one of a group of alternatives (a character class).

For example, a search for:

AUTHOR matches jo[ah]n

matches records with AUTHOR fields containing "Johnson, Stephen M.," "Narciso, John, |d1924-," and "Fromer, Margot Joan, |d1939-."

You can **use a hyphen** within square brackets to specify a range of characters, for example, "[0-9]" matches any digit and "[a-z]" matches any letter. You can also combine the square brackets with the plus sign (or the asterisk) to search for a string of characters of any length:

AUTHOR matches [a-z] man

matches records with AUTHOR fields containing "Maurer, Hermann A.," "Rothman, Stanley," and "Enger, Norman L."

If the first character in the square brackets is a **caret '^'**, **then the search matches all characters that are not in the character class.** See About Subfield Tags for an example of this.

{ } **Curly braces** allow you to specify that a character or pattern must occur a specific number of times.

For example:

TITLE matches l{3}

matches records with TITLE fields containing the letter 'l' appearing three times in succession, such as "illustrated". This can be useful in catching data entry errors in the database.

( ) **Parentheses** create groups that can be followed by modifiers.

For example:

TITLE matches (the ){2}

matches records with TITLE fields containing the word "the" appearing twice in succession, such as "The the book of lists". This can be useful in catching data entry errors in the database.

You can combine these basic regular expression symbols to create very sophisticated search patterns. Consult a book on the UNIX egrep program to learn about other regular expression symbols and search techniques.