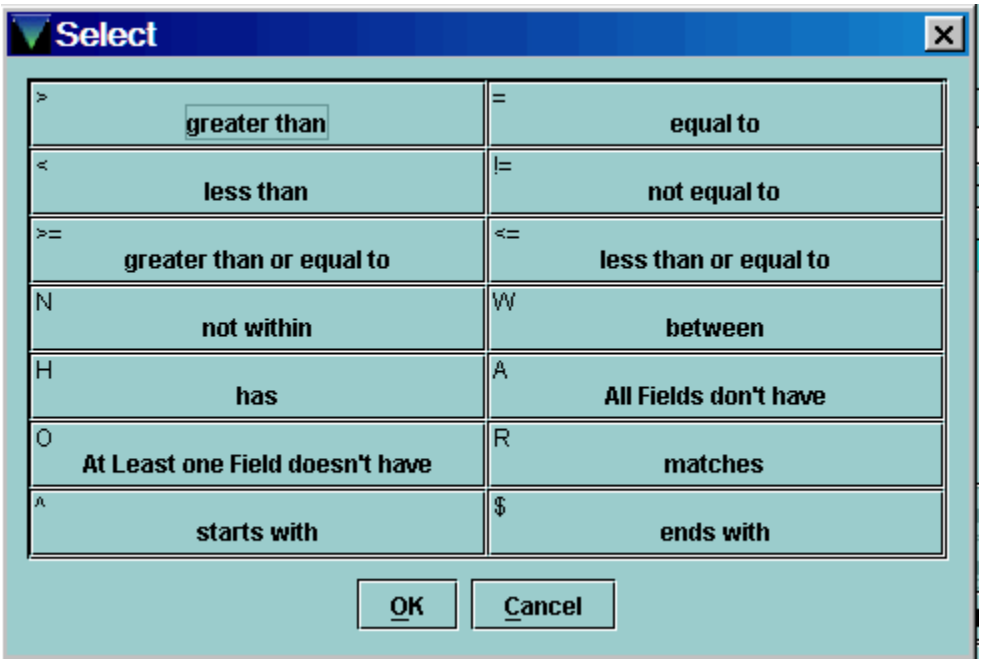
BOOLEAN OPERATOR HELP SHEET
FOR CREATING LISTS

When you create Boolean search criteria for a review file, you need to be very familiar with the BOOLEAN CONDITIONS AVAILABLE. With Millennium Silver, the conditions Starts With and Ends With have been added.



The Boolean conditions are terms or symbols that tell the system the relationship between the fields that you want to search and the data in that field.

= equal to

Data in the record must *exactly match* the data in the query. This is best for **fixed fields** and usually not recommended for variable fields.

- ✓ Useful for things like '**MAT TYPE = m**' (for software)
- ✓ Lib=miad (system won't find any items that MIAD and other SWITCH libraries hold…strictly MIAD's items (has would be a better choice)

You can also use the equals sign to find all the records that do NOT contain a particular **variable field**. You choose the field name, use the equals sign and leave the value space blank. This search means *find records for which that field equals nothing*. You could use this to find patrons who do not have an e-mail address in their record, or items that do not have a call number. However, it will also find records that have that field with nothing in it (i.e. the field exists but is blank).

!= not equal to [~ symbol in text-based INNOPAC and turns into <>)

This works like the equals sign, only it **EXCLUDES** rather than includes records that match the criteria you specify. (Data does NOT exactly match data in the query). It's very useful for **fixed fields**. You can use it as part of a longer search string to exclude some group-- you might want a list of all patrons who have owe a certain amount of money, except for faculty, or you might want of all outstanding orders, except for ones from one particular fund.

> **Greater than**
< **Less than**
>= **Greater than or equals to (g in text-based INNOPAC)**
<= **Less than or equals to (l in text-based INNOPAC)**

These are mostly used for **fixed fields containing numerical data, like dates and money**, and work exactly the way you would expect them to. The only tricky part here is making sure you are using the right one. Be careful when using either Less Than operator-- remember that a blank field is going to be Less Than whatever value you specify. If you search for DVDs whose due date is less than 01-01-2002, you'll get all the items with a blank due date, because they are not checked out. *Price is less than $25* will find record with a blank price field. A date is "greater than" the date specified if it comes later. An earlier date is "less than" the specified date.

These Boolean operators perform a character-by-character comparison between the search statement and the data in the specified field, starting with **the leftmost character** in each. INNOPAC stops the comparison when the first non-matching character is reached.

>= and <= will retrieve exact matches too. If an item has an exact price of $50, a search for **"PRICE >= 50"** will retrieve that record, as will a search for "**PRICE <=50"**

W **Between (or within)**
N **Not within**

Important operators. You can use W or N to specify a range, usually used for **fixed fields containing dates, amounts or numbers**. These are the only operators that require a second value.

- ✓ **"PRICE between 25 & 50"** to locate records with prices between $25 and $50
- ✓ "**BARCODE between 1110000000000 & 1119999999999"**

It can also be used for specifying alphabetical ranges. However, you have to be very careful in selecting the values. For example, "**LOCATION between m & mz**" will include locations m, ma, mcav, and mz, but **not** location mza. "**LOCATION not within m & mz**" will include locations a, lzzzz, n, and zzz. It will **also** include locations mza through mzzzz.

H **Has**

Important operator. Good for **variable fields** and searching unique character strings. A record will be retrieved if data is contained **as a group anywhere in the specified field**; the entire group of characters must match correctly though, including spaces and punctuation. (e.g., "**loc has shst**")

A **All Fields Don't Have (X in INNOPAC for 'does not have')**
O **At Least One Field Doesn't Have (is a prompted question in INNOPAC)**

These two operators are used to find records that **DO NOT HAVE** a certain string of characters in a field. The second one (at least one field doesn't have) is only needed in situations where there may be **multiple variable fields with the same tag in the same record**, and you want to make sure that every one of those fields has some particular string of text.

R Matches Regular Expression

This works like Has, but allows you to do **pattern matching using UNIX regular expressions**. This can get complicated, but the most common uses are simple. You can specify two or more characters in square brackets, and the system will find records that match any of them. For example: **Title matches gr[ae]y ghost** will find titles that have either *grey ghost* or *gray ghost*. You can also specify a range, like [0-9] or [a-z]. These are useful in finding patterns like University ID numbers, dates, etc. For example 1[8-9][0-9][0-9] would find any four digit number from 1800-1999. See Richard V. Jackson's handout from his presentation on regular expressions at IUG13.
http://innovativeusers.org/iug2005/programs/materials/iug13G4.pdf (requires IUG user name and password)

^ Starts with

New with Millennium Silver, this operator fiends field which start with the specified value. MARC tags and indicators and |a at the beginning of a field are ignored.

✓ "**BARCODE starts with 111"**

$ Ends with

New with Millennium Silver, this operator fiends field which end with the specified value. Any punctuation at the end of the field must be included.