Mastering Create Lists

Richard V. Jackson

The Huntington Library, Art Collections, and Botanical Gardens

San Marino, California

The Huntington Library

The Huntington Library

Outline of the presentation

1) Review of the fundamentals

2) Enhanced queries in Sierra

3) Working with Create Lists

4) Using regular expressions

5) JSON and the ‘in’ operator

Questions, problems, discussion anytime

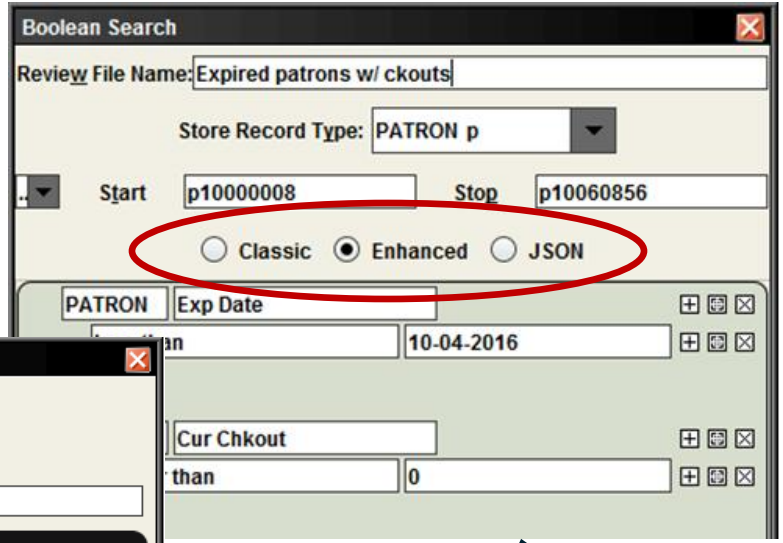
Why this presentation?

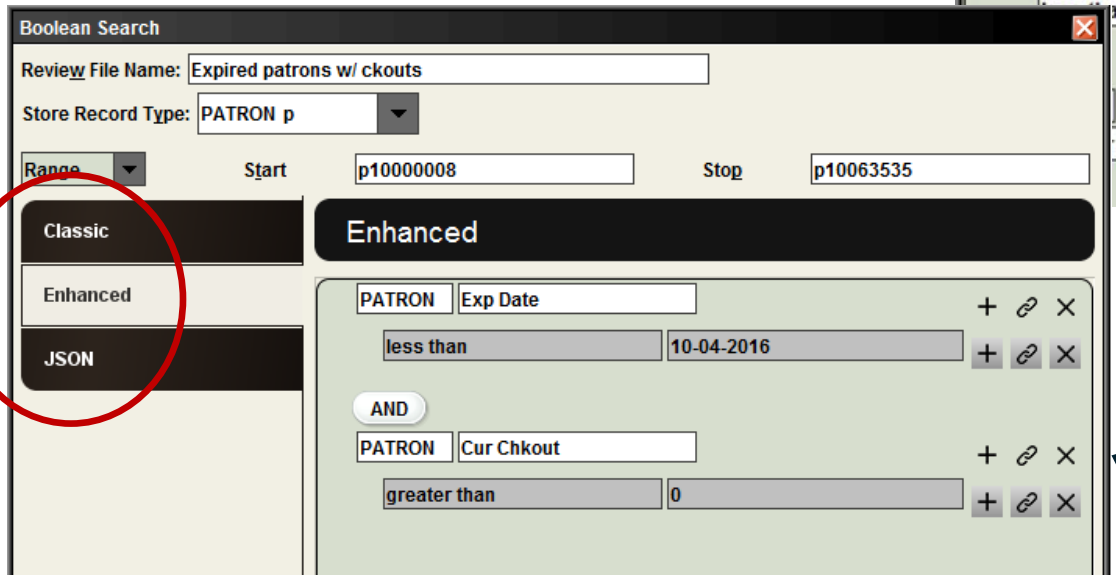
- Create Lists is an essential tool
- Integrated with many functions
- Easier to learn than SQL
- New methods and functionality added in recent Sierra releases

Testing for this presentation done in Sierra 3.4

New query builders

- With Sierra release 2.2, “Enhanced” and “JSON” query builders added to “Classic”





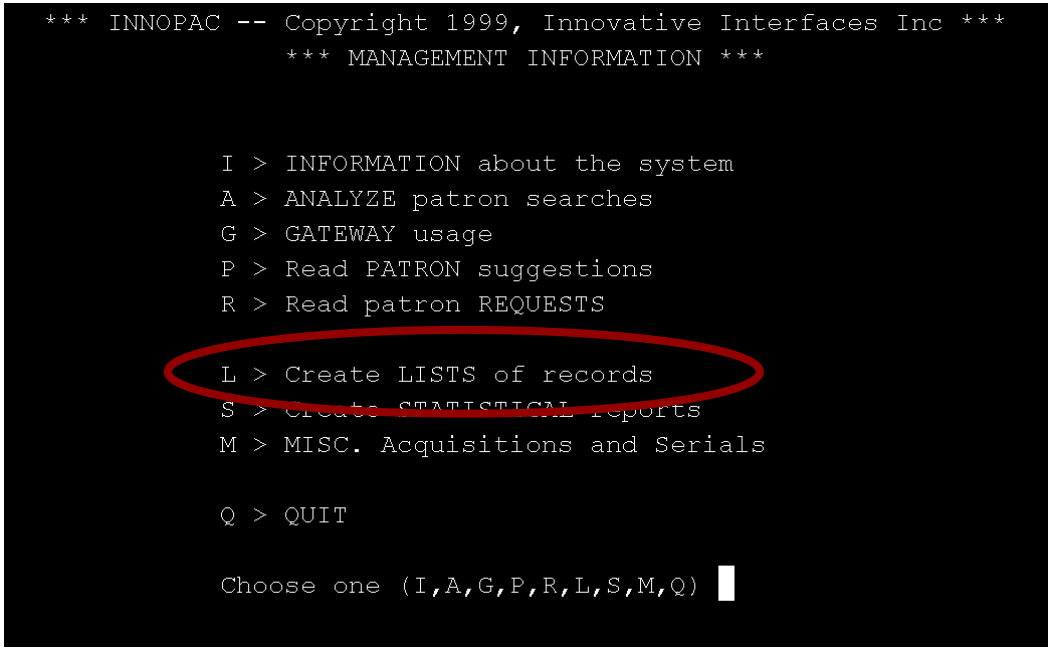
Sierra 2.2 interface

Sierra 2.4+ interface

1) Review of the fundamentals

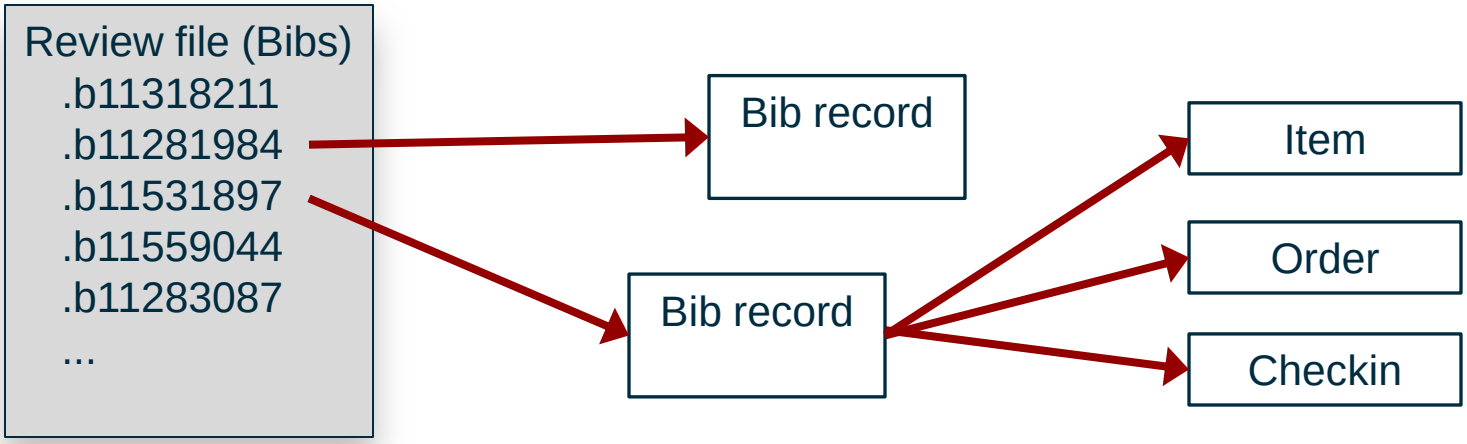
What is “Create Lists”?

- Tool that stores selected records of any type in a review file
- Uses a sophisticated but easy-to-learn query interface
- Enables you to extract records that share any combination of attributes you can define, including attributes in linked records



What is a review file?

- List of pointers, i.e., record numbers
- Records viewed or edited in a review file are the actual records, not “snapshot” copies of the records
- Stores only one type of record (but linked records can also be viewed and edited within Create Lists)



Permissions for Create Lists

Permission 018 (Create lists)

- to create, view, and maintain review files
- must also be able to view or edit records

Permission 186 (Review file list administrator)

- to change ownership of review files

Permission 272 (Advanced System Administration)

- to change the number and size of available review files (must also have permission 186)

Permissions: Create Lists templates

- Enable in Database Maintenance options (Admin Corner):

```
38 > Create BOOLEAN lists: use templates files..............YES
```

Permission 640 (Create Lists Limited)

- cannot create new queries or alter search strategies
- must use existing search templates, changing only the values as needed

Permission 641 (Create Lists Template Admin)

- to create, edit, and delete search templates in Create Lists (must also have permissions 18 and 186)

Advantages to Create Lists

- You get a review file!
  - Persists until emptied
  - Accessible in other modules/workflows, by other people
  - Sort, add and remove records
- Not dependent on indexing
  - “Get at” virtually all parts of a record
  - Search for presence or absence of fields
- Complex Boolean searches
  - Search combinations of multiple attributes
  - Use attributes in attached records

Create Lists queries vs. index searches

- Queries in Create Lists not dependent on indexing
  - “Brute-force” retrieval
  - In general, searches data as they are stored
- Index entries are normalized
  - Punctuation, subfield delimiters ignored or converted to space
  - Characters with diacritics converted to corresponding plain text
  - Ampersand (&) converted to “and”
  - Initial articles in titles may be ignored (when field uses a non-filing indicator)

Create Lists queries vs. index searches

```
245 10 Lincoln :|ba novel /|cGore Vidal.
```

A regular index search:

will retrieve the above

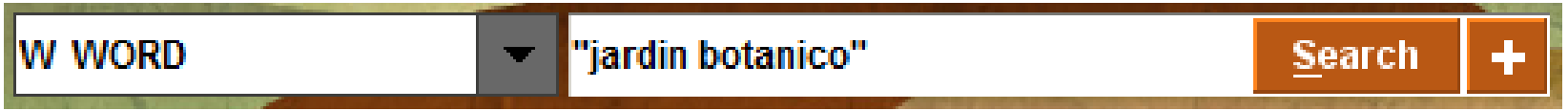
A similar query in Create Lists:

Type	Field	Condition	Value A
BIBLIOGRAPHIC	TITLE	starts with	lincoln a novel

will not retrieve it

Create Lists queries vs. index searches

This keyword search:



will retrieve a record with this field:

```
500  "To celebrate the inauguration of the Jardín
     Botánico de Córdoba."
```

This query in Create Lists will *not* retrieve it:

Type	Field	Condition	Value A
BIBLIOGRAPHIC	NOTE	has	jardin botanico

Letter case normalization

- Letter case normalization ***does*** apply in Create Lists

Create Lists query:

```
Bib Title has "VISTA"
```

Data from records:

```
Monte Vista! :|b
something new and...
```

```
The politics of the
Peace Corps & VISTA
```

```
Rivista geograpfica
italiana
```

Letter case normalization

- Letter case normalization ***does*** apply in Create Lists
- When a search is run, data in both records and queries are changed to lower case

Create Lists query:

```
Bib Title has "vista"
```

Data from records:

```
monte vista! :|b
something new and...
```

```
the politics of the
peace corps & vista
```

```
rivista geograpfica
italiana
```

- Result: Searches are case-insensitive

Changing the base range for searching

- You can change the record number range to be searched:

Use “i*” (or “b*”, “p*”, etc.) to indicate the highest record #

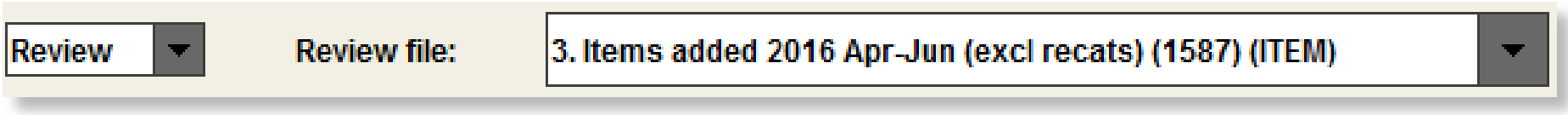
- Or search within an index range:

Note: In Sierra, this range covers the beginning of the Q’s through the end of the R’s; in Millennium only the Q’s are retrieved (the first ‘R’ call number would be greater than ‘R’)

Changing the base range for searching

- Search within the results of an advanced keyword search:

- Or search within another review file:

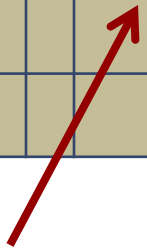


You can store a different record type than the records in the review file being searched (e.g., make a list of items from a list of bibs)

The “Classic” query builder

Store Record Type: BIBLIOGRAPHIC b ▼

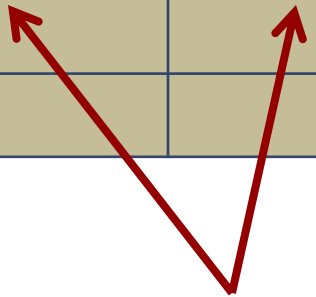
Operator		Type	Field	Condition	Value A	Value B	
		BIB	CAT DATE	between	01-01-2017	03-31-2017	



Record type
[target]

Field
[field identifier]

Condition
[operator]



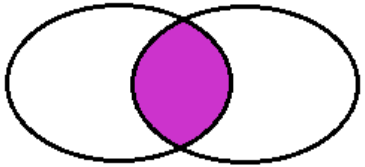
Value(s)
[operands]

The “Classic” query builder

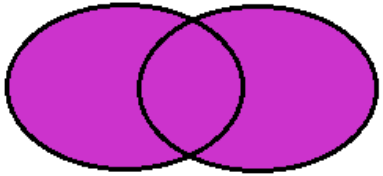
Store Record Type: **BIBLIOGRAPHIC b** ▼

Operator		Type	Field	Condition	Value A	Value B	
		BIB	CAT DATE	between	01-01-2017	03-31-2017	
AND		ITEM	LOCATION	equal to	gccb		

Boolean operator



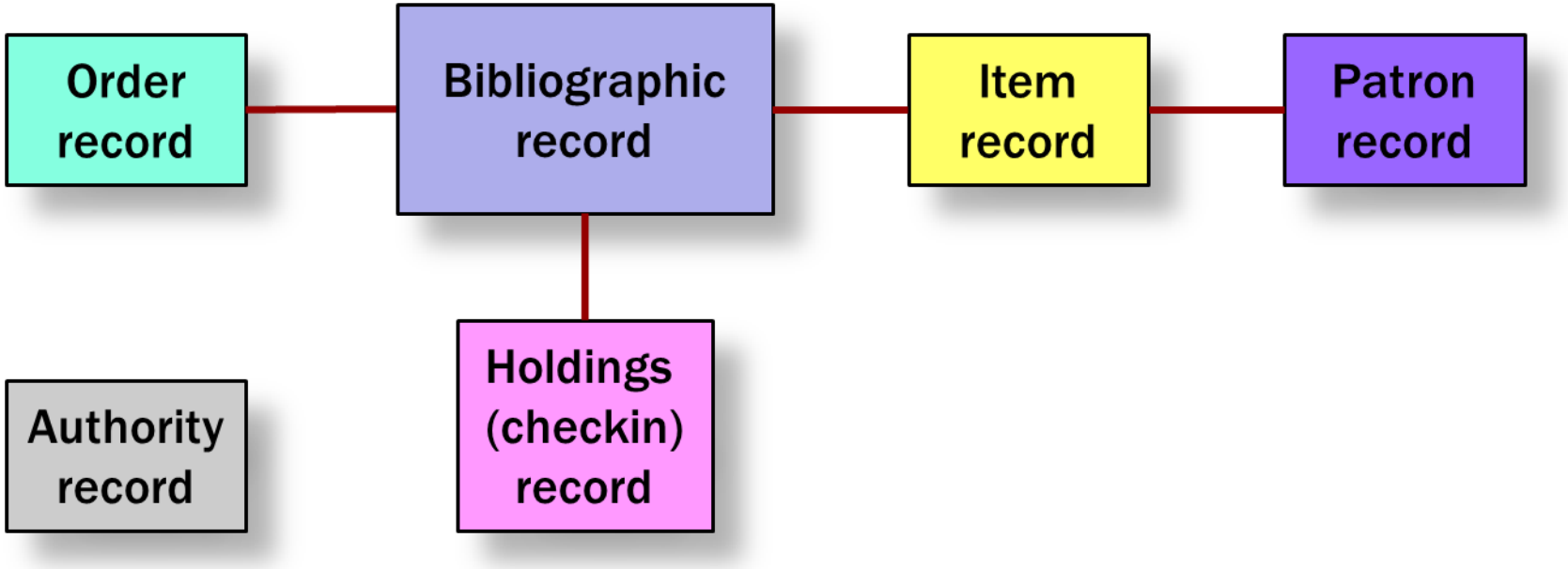
AND (both statements must be true)



OR (either statement may be true)

Rule of “one hop”

Queries may search fields in the record being stored, and fields in attached records no more than one link away



Complex queries

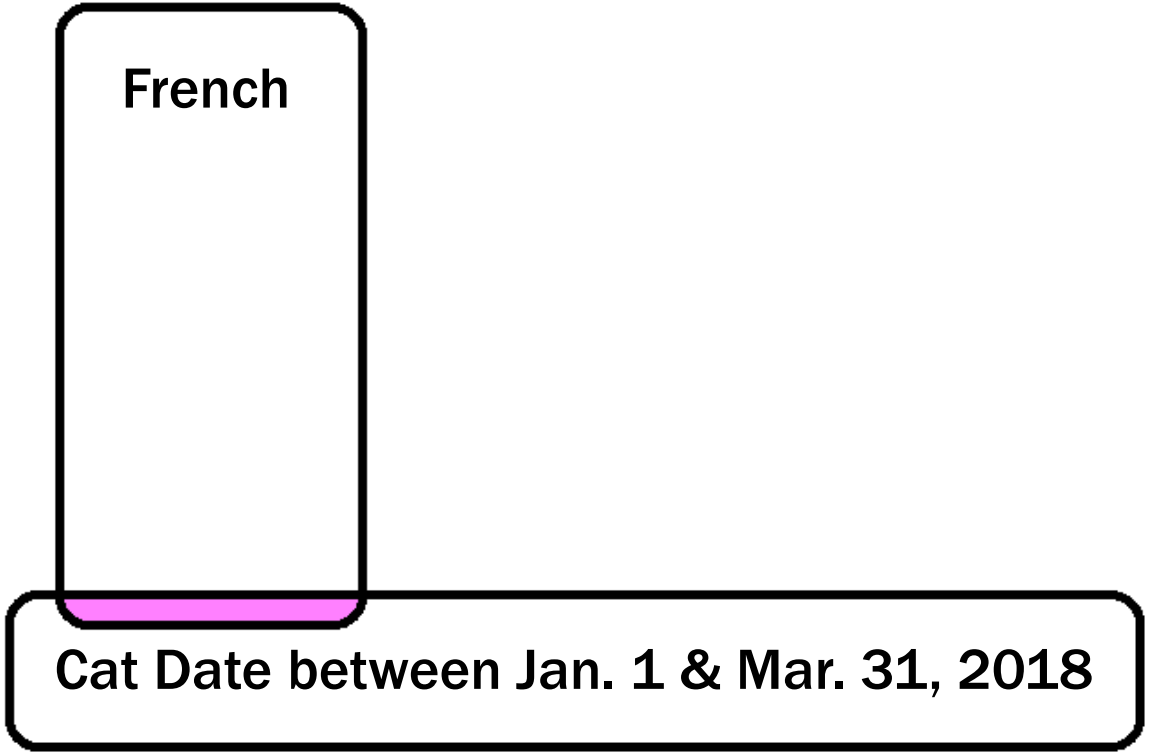
Operator		Type	Field	Condition	Value A	Value B	
		BIB	CAT DATE	between	01-01-2018	03-31-2018	
AND		BIB	LANG	equal to	fre		
OR		BIB	LANG	equal to	ita		
OR		BIB	LANG	equal to	spa		

Too many records found!

When 3 or more search statements mix AND and OR, you may need to **group** them to control the order in which statements are evaluated

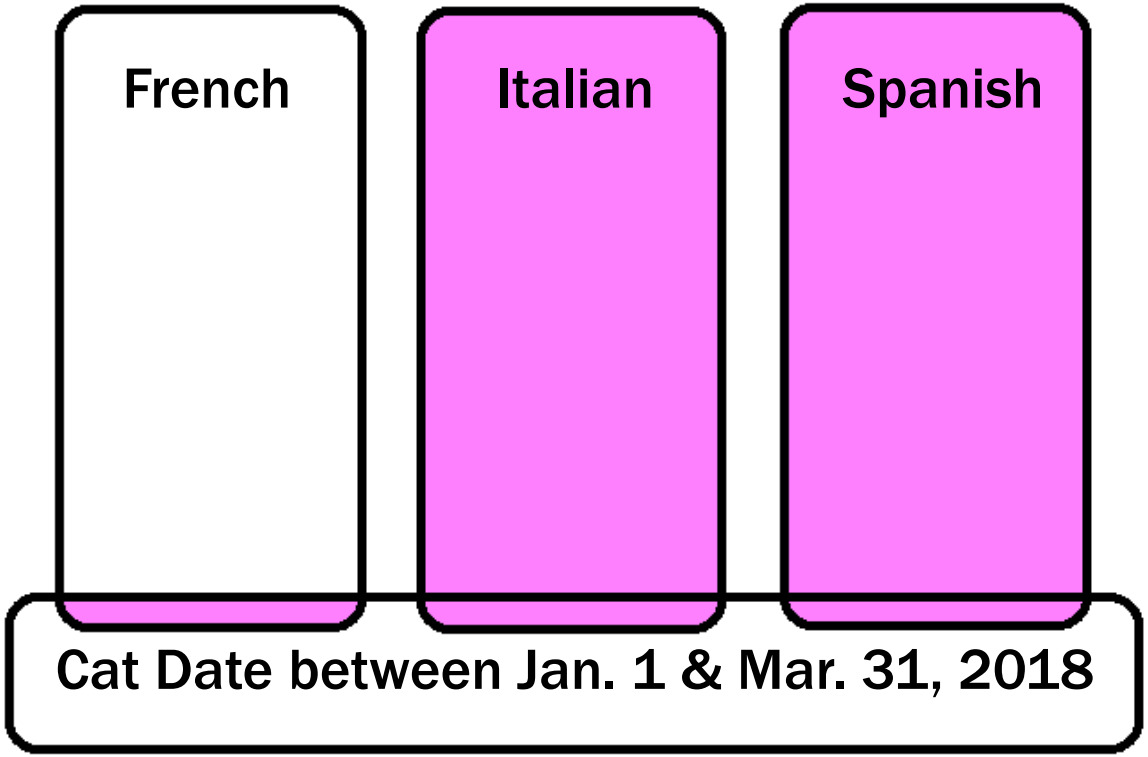
Complex queries

The first 2 statements – connected by AND - are evaluated first



Complex queries

The results of this operation are then OR'd with the next statement, and then the next



Complex queries: grouping

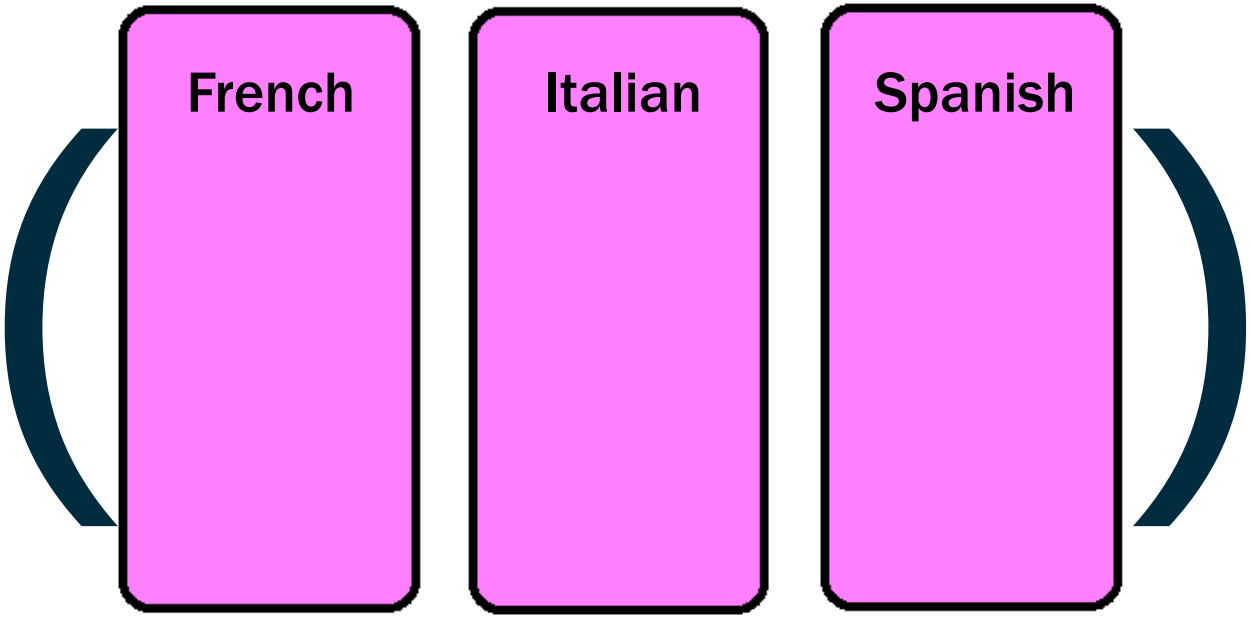
Operator		Type	Field	Condition	Value A	Value B	
		BIB	CAT DATE	between	01-01-2018	03-31-2018	
AND	(	BIB	LANG	equal to	fre		
OR		BIB	LANG	equal to	ita		
OR		BIB	LANG	equal to	spa		)

To fix this, group the statements to be evaluated first

In the Classic query builder, click and drag across two or more statements, then click the **Group** button

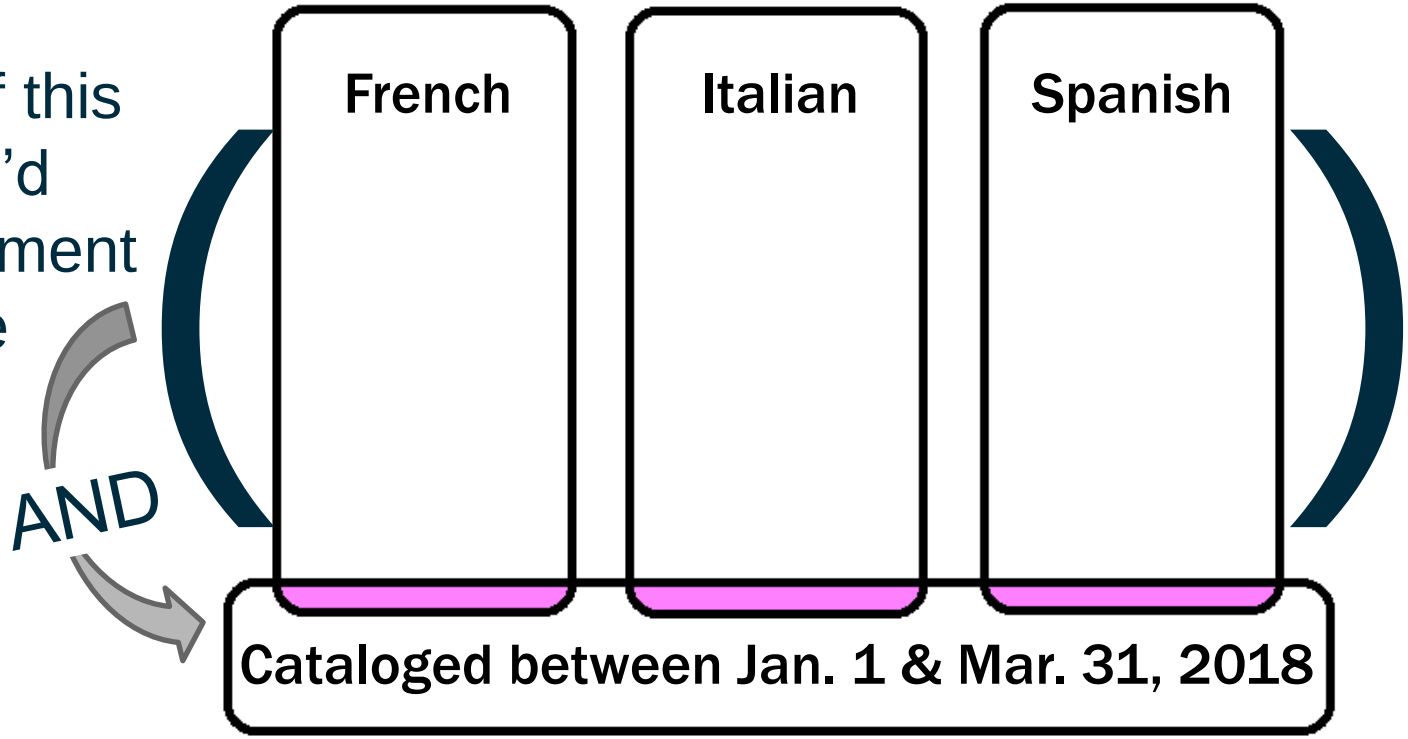
Complex queries: grouping

With grouping, the 3 language statements are first OR’d together



Complex queries: grouping

The results of this are then AND'd with the statement specifying the Cat Dates



Queries: specifying fields

- Fields to be queried can be specified as:
  - A variable-length field

Type	Field
ITEM	VOLUME

  - A fixed-length field

Type	Field
PATRON	Patron Type

  - A special field (component of 006, 007, 008)

Type	Field
BIBLIOGRAPHIC	Date Two

Double-click the input box to see a menu of available fields

Queries: specifying fields

- Fields can also be specified as a MARC tag

  (Type “!” to call up the MARC Field input box)

`245`	MARC tag 245	`5??`	All 5xx fields
`24514`	245 with indicators 14	`26?`	Any 26x field
`245|ab`	245 subfields a & b	`6???7`	6xx fields with 2nd ind. 7

Queries: specifying fields

- Fields can also be specified as a MARC tag

  (Type "!" to call up the MARC Field input box)

245 MARC

24514 245 wi

245|ab 245 subfields a & b

Oops—this no longer works in Sierra 3.4.
It *was* working in releases at least through Sierra 3.1. (I'm not sure about 3.2-3.3.)
However, there are other ways to do this.

6???7 6xx fields with 2nd ind. 7

Queries: Conditions (operators)

Operator	Keyboard input
equal to	=
not equal to	!=
greater than	>
less than	<
greater than or equal to	>=
less than or equal to	<=
between	w

Operator	Keyboard input
not within	n
has	h
all fields don't have	a
at least one field doesn't have	o
matches	r
starts with	^
ends with	$

Queries: Equal to / Not equal to

- Generally, don’t use ‘equal to’ and ‘not equal to’ with variable-length fields

  BIB TITLE equal to “moby dick’ will not find:

  245 10 |aMoby Dick, or, The white whale /|cby Herman Melville.

  It won’t even find:

  245 10 |aMoby Dick.

- Use ‘has’ (or ‘starts with’ or ‘ends with’) instead

Queries: Equal to / Not equal to

- Generally, don't use 'equal to' and 'not equal to' with variable-length fields
- Exception: Searching for the presence or absence of a field

```
PATRON   EMAIL ADDR   equal to   ""
```

Finds Patron records with no email address

```
BIB   245|h   not equal to   ""
```

Finds Bib records where the 245 field contains a subfield h

Queries: new operators for *DATE* fields

- Available in Sierra 2.0+
- Used only with date fields
  - Created
  - Updated
  - Bib: Cat Date
  - Item: Due Date
  - Patron: Exp. Date *[etc.]*
- Relative dates especially useful with Scheduler

Operator	Keyboard input
exist	e
not exist	n
equals today	t
equals yesterday	y
within last week	v
within last month	m
is this many days ago	a
is this many weeks ago	b
is this many months ago	c

Queries: tips for using date operators

- No values (operands) are needed with date operators — except for "*is this many* [*days*|*weeks*|*months*] *ago*"
- A "*less than*" (<) condition with a date field does not include blank dates (in Sierra)
- "*within last month*" — means from the first through the last day of the most recent whole month
- "*is this many weeks ago*", "*is this many months ago*" — refers to the ***one day*** occurring exactly *x* number of weeks or months ago

“Linked record” searches

- Checks for the presence or absence of linked records
- Works only with the BIBLIOGRAPHIC record type
- Keyboard input for “linked record” field: **^**

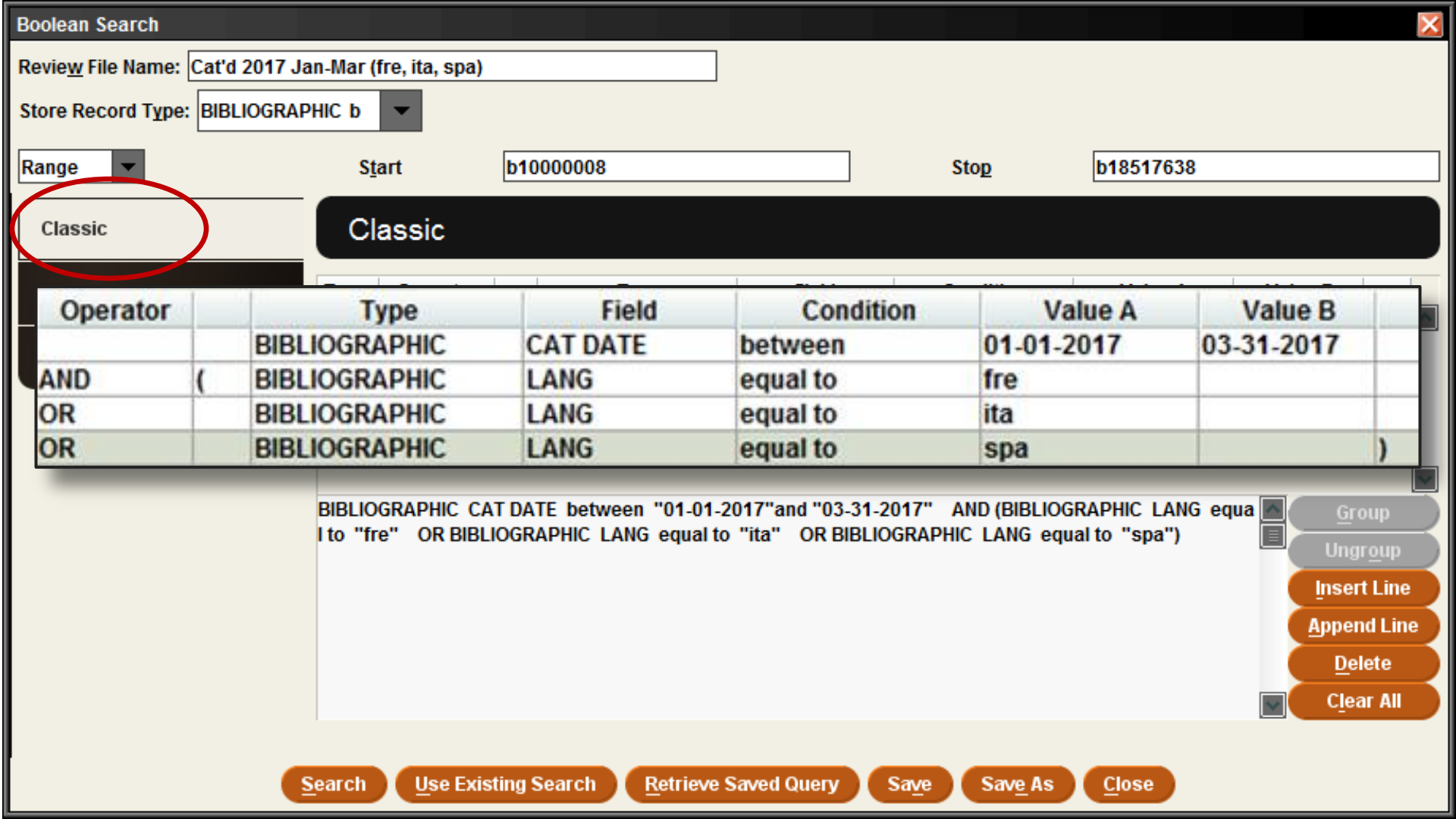
Operator		Type	Field	Condition	Value A	Value B	
		BIB	LINKED RECORD	not exist to	ITEM		
AND		BIB	LINKED RECORD	exists to	ORDER		

- Condition can only be ‘exists to’ (**e**) or ‘not exist to’ (**n**)
- Value can only be ‘ITEM’ (**i**), ‘ORDER’ (**o**), or CHECKIN (**c**)

2) Enhanced queries in Sierra

Introducing the Enhanced query builder

- “Next generation” Create Lists introduced with Sierra 2.2
- More intuitive query interface (maybe)
- Clearer visual representation of the query
- Gets around limitations of the table-format used by the “Classic” query builder
  - Can have multiple terms applying to one field
  - Can have more than two operands (values)


Boolean Search
Review File Name: Cat'd 2017 Jan-Mar (fre, ita, spa)
Store Record Type: BIBLIOGRAPHIC b
Range
Start
b10000008
Stop
b18517638
Classic
Classic
Operator | Type | Field | Condition | Value A | Value B
BIBLIOGRAPHIC | CAT DATE | between | 01-01-2017 | 03-31-2017
AND (| BIBLIOGRAPHIC | LANG | equal to | fre
OR | BIBLIOGRAPHIC | LANG | equal to | ita
OR | BIBLIOGRAPHIC | LANG | equal to | spa)
BIBLIOGRAPHIC CAT DATE between "01-01-2017"and "03-31-2017" AND (BIBLIOGRAPHIC LANG equa l to "fre" OR BIBLIOGRAPHIC LANG equal to "ita" OR BIBLIOGRAPHIC LANG equal to "spa")
Group
Ungroup
Insert Line
Append Line
Delete
Clear All
Search
Use Existing Search
Retrieve Saved Query
Save
Save As
Close

Classic

Enhanced

JSON

Enhanced

BIBLIOGRAPHIC | CAT DATE

between | 01-01-2017 and 03-31-2017

AND

BIBLIOGRAPHIC | LANG

equal to | French

OR

BIBLIOGRAPHIC | LANG

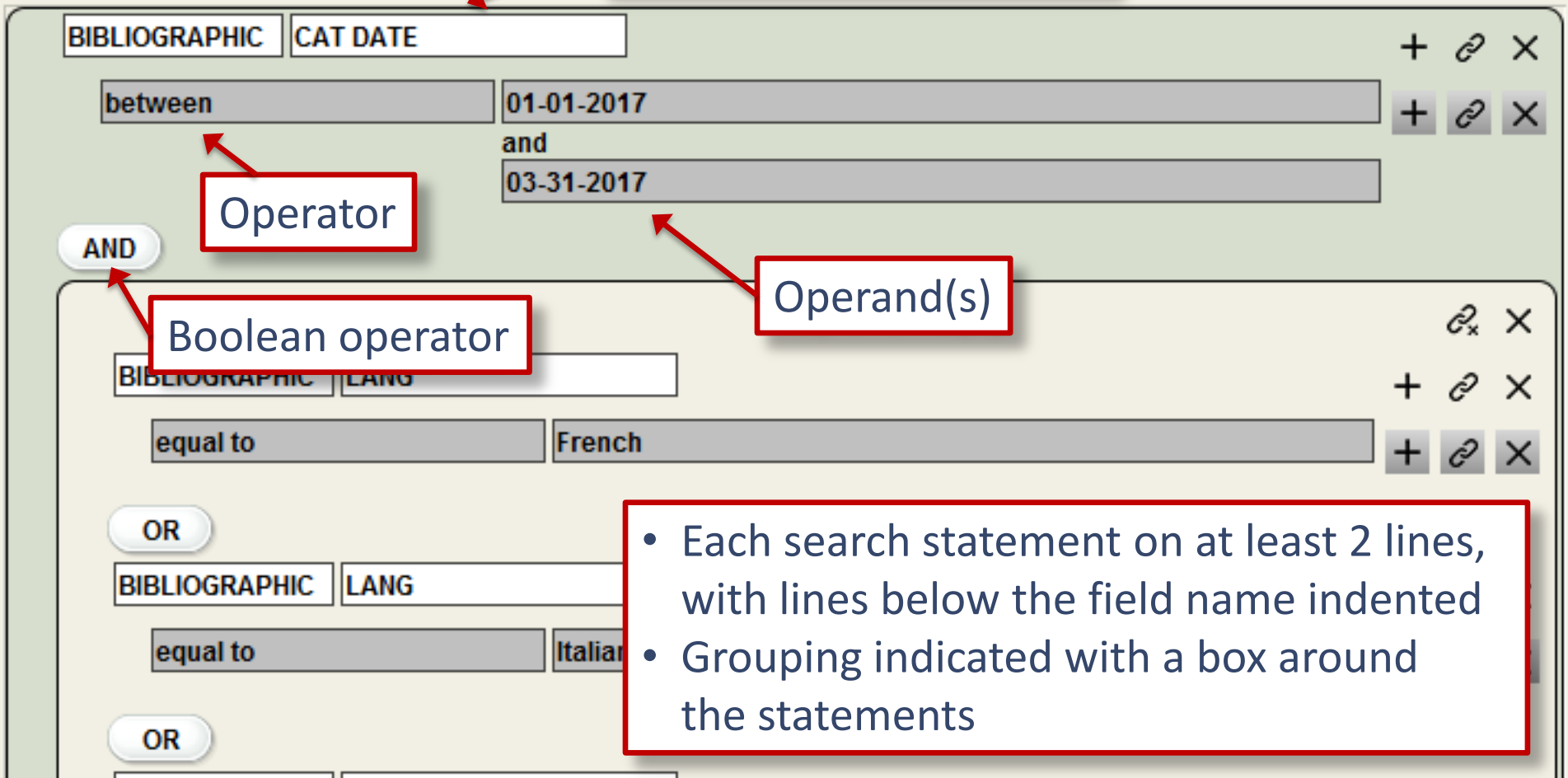
equal to | Italian

OR

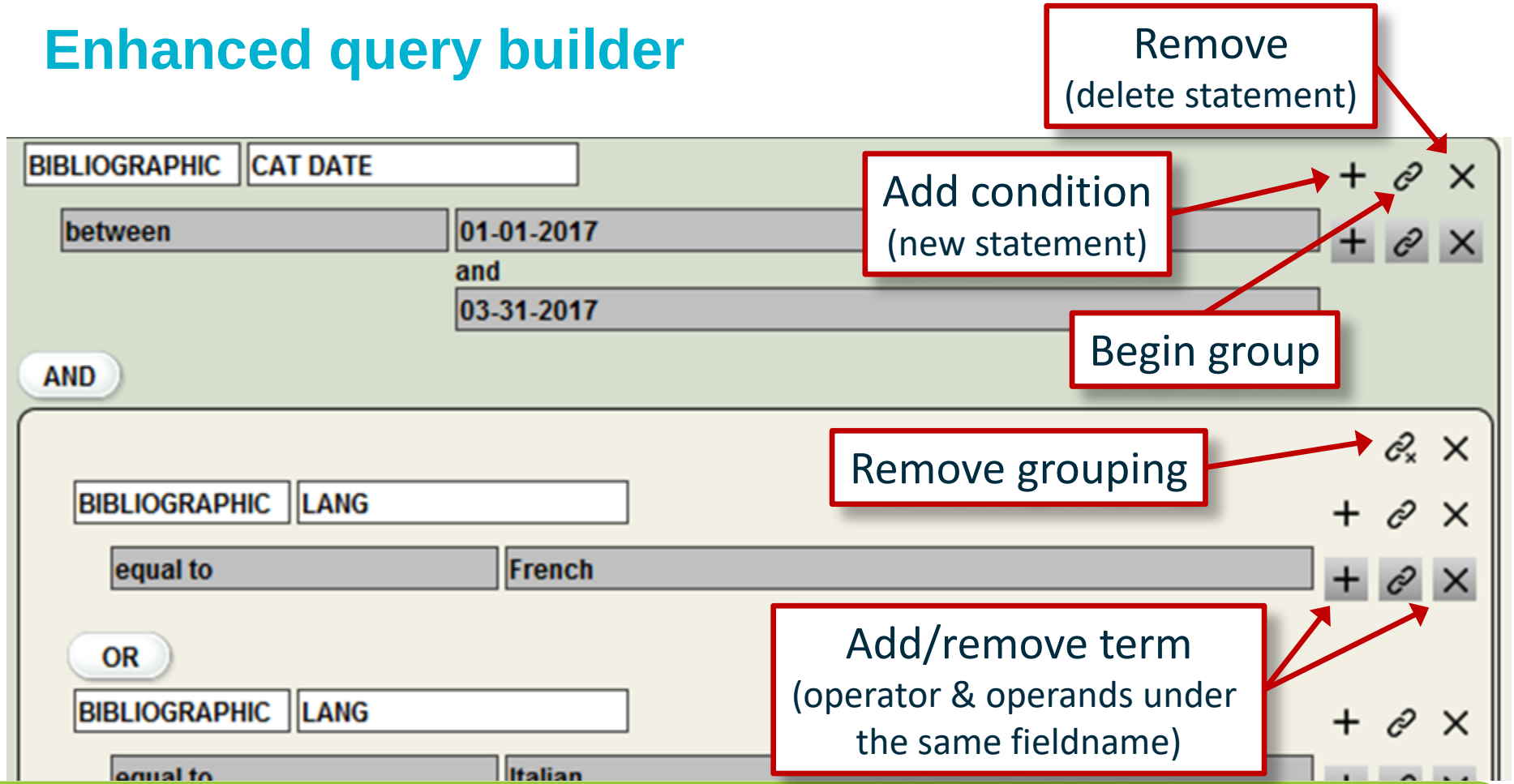
BIBLIOGRAPHIC | LANG

equal to | Spanish

Enhanced

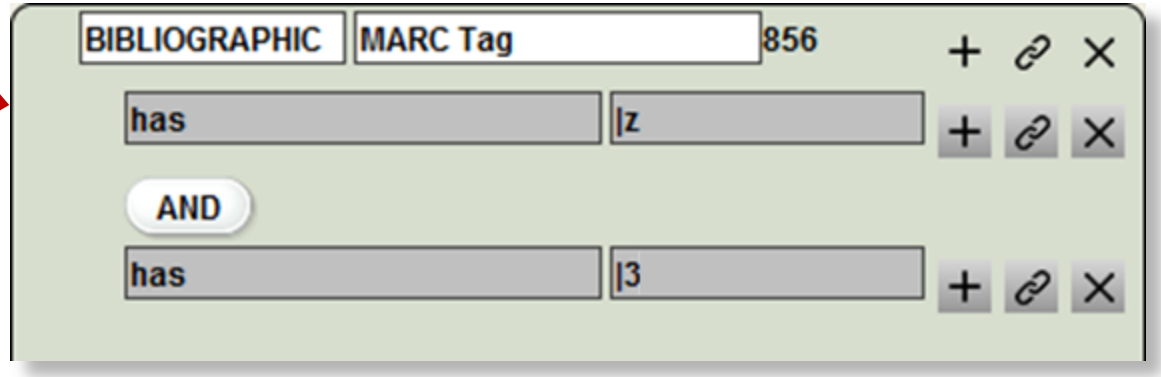

Record type, Field name
BIBLIOGRAPHIC
CAT DATE
between
01-01-2017
and
03-31-2017
Operator
AND
Boolean operator
Operand(s)
equal to
French
OR
BIBLIOGRAPHIC
LANG
equal to
OR
• Each search statement on at least 2 lines, with lines below the field name indented
• Grouping indicated with a box around the statements

Enhanced query builder



Multiple terms for a single target (field)

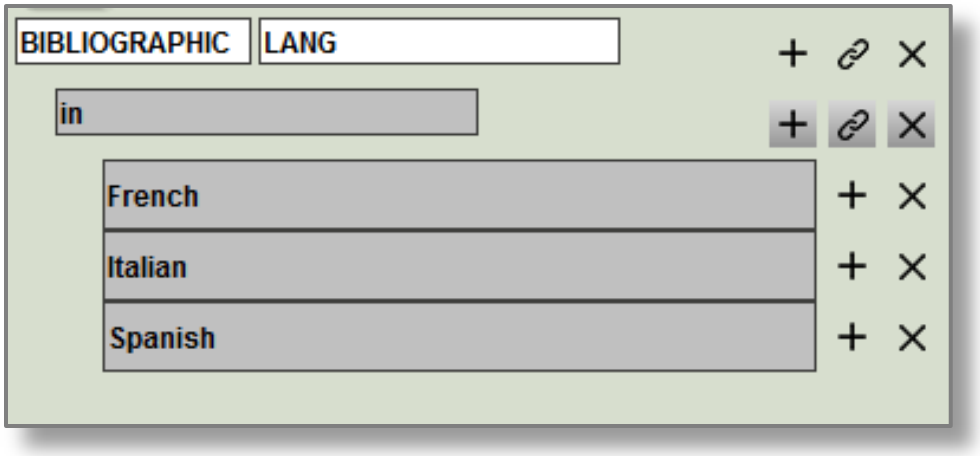
- Evaluated as a single search statement — terms are automatically “grouped” (evaluated first)
- Multiple terms must apply to the same instance of the field
  - In this search, one 856 must contain both “|z” and “|3”
  - This search may find records where “|z” and “|3” occur in different 856s



Term	Operator		Type	Field	Condition	Value A
1			BIBLIOGRAPHIC	MARC Tag 856	has	|z
2	AND		BIBLIOGRAPHIC	MARC Tag 856	has	|3

The “in” operator

- The only operator that can take more than 2 operands
- The logic: “Is the value of the field being searched ***equal to*** one of the members of the set of values listed here?”
- Works best with fixed-length fields or simple variable-length fields (barcodes, OCLC nos.)



3) Working with Create Lists

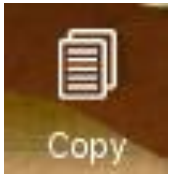
System-generated review files

- The system automatically maintains some review files
  - OVERDUE Items
  - Items with HOLDS
  - Items on Holdshelf *[etc.]*
- Other review files can be created as part of some process
  - Record loading through Data Exchange 

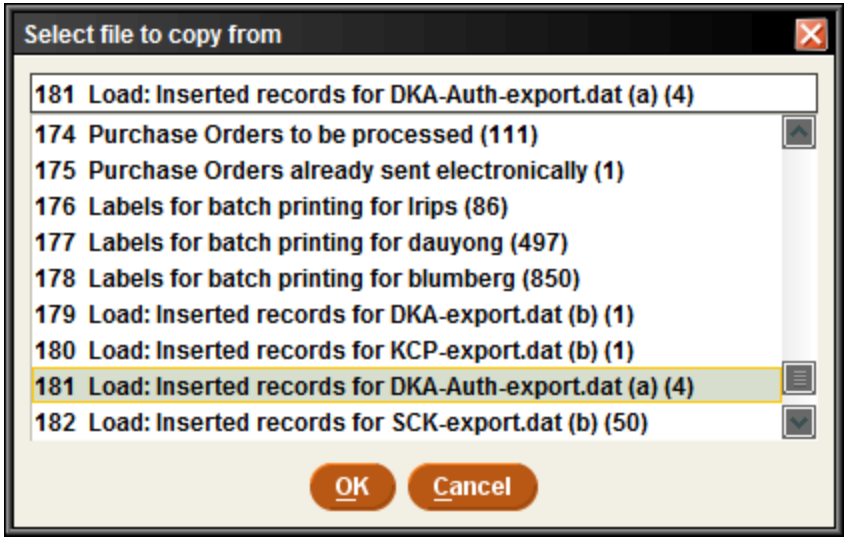
  - Queue monographic label to print (but not when using print templates)
  - Copy records in a Headings Report to a review file (limited usefulness)

System-generated review files

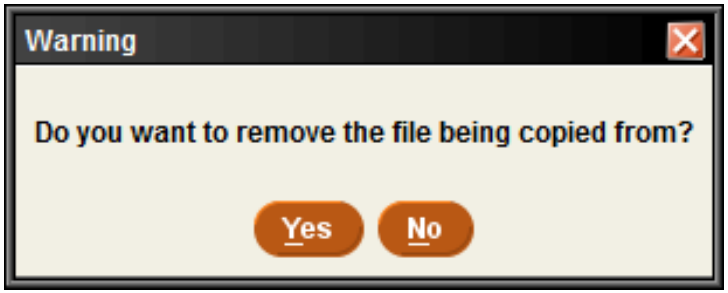
To view a system-generated review file:



- Select an empty review file and click the Copy button
- Scroll to the bottom of the review file list and select the file you want



- When asked:



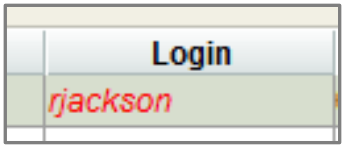
generally say “Yes.”

Owning and releasing review files

- Enable this feature in Admin Corner (A > A > S > O > D):

```
27 > Create BOOLEAN lists: password review files.............YES
```

- Clicking “Own” for a selected review file means:
  - Only that Login can empty or modify that review file
  - Only that Login can “Release” ownership of the file
  - Owners login name appears in red italics
  - Note that other logins can still open the review file and view or edit the records in it



- List Administrators (permission 186) can override ownership

Repeating a previous (but unsaved) query

1. Select a review file and click 

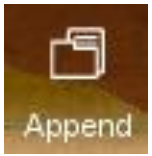
2. At the bottom, click 

3. Select the review file whose search strategy you want to copy
4. Modify the search strategy as needed
5. If desired, save the query before executing it 

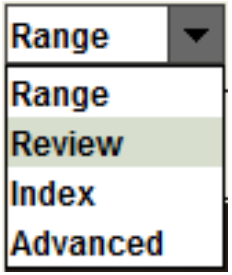
Useful for correcting a mistake in a query without having to reenter the whole thing, or to save a query that had already been executed.

Merging two review files

1. Select the first review file and click “Append”



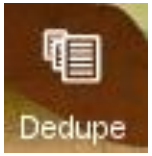
2. Change “Range” to “Review” in the drop-down menu and select the second review file



3. Do not enter any search statements

4. Click “Search”

5. Click “Dedupe” to remove duplicate records

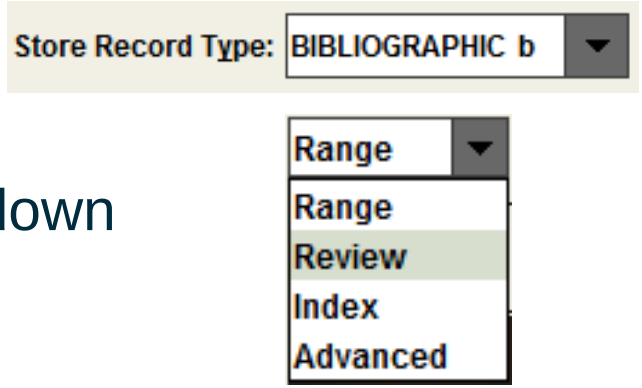


6. If it’s no longer needed, empty the second review file

Changing the record type of a file

For example, you have a review file of items, but you need to use Global Update to change the bib records they're linked to

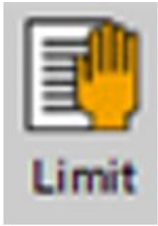
1. Select an empty review file, click Search, and choose the Bibliographic record type
2. Change "Range" to "Review" in the drop-down menu and select the review file of items
3. Execute the search, then dedupe the file



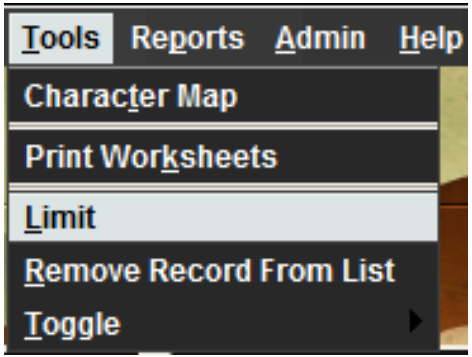
This can be done in reverse (bibs to items), or between any directly linked record types (e.g. items ↔ patrons, orders ↔ bibs, etc.

Using review files in Global Update

- I prefer to use Global Update on records in a review file (rather than using an index or advanced keyword search)
- It's safer—if something goes wrong, you still have the records in a review file
- Global Update has a Limit function that works like the Classic query builder in Create Lists
  - In Millennium, there is a Limit button



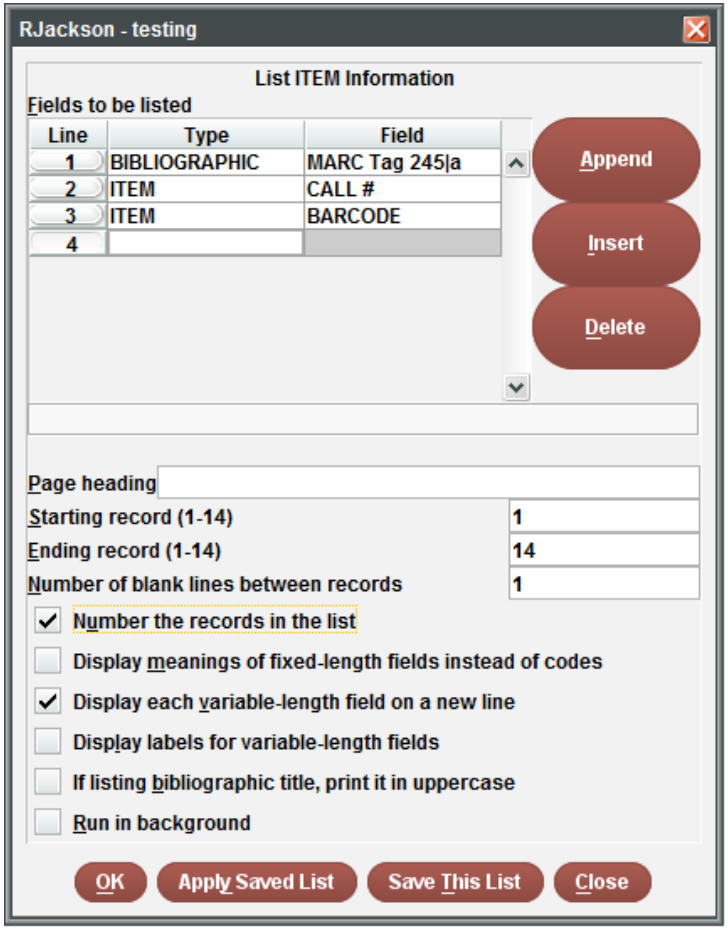
  - In Sierra , it's in the Tools menu



Listing records

List Records

- Designed for printing out selected fields from a review file
- Formatting is very limited — but you can print to email and then paste into Word
- The listing information (fields to be included, etc.) can be saved for future use. (That's what the "Saved Lists" tab refers to.)

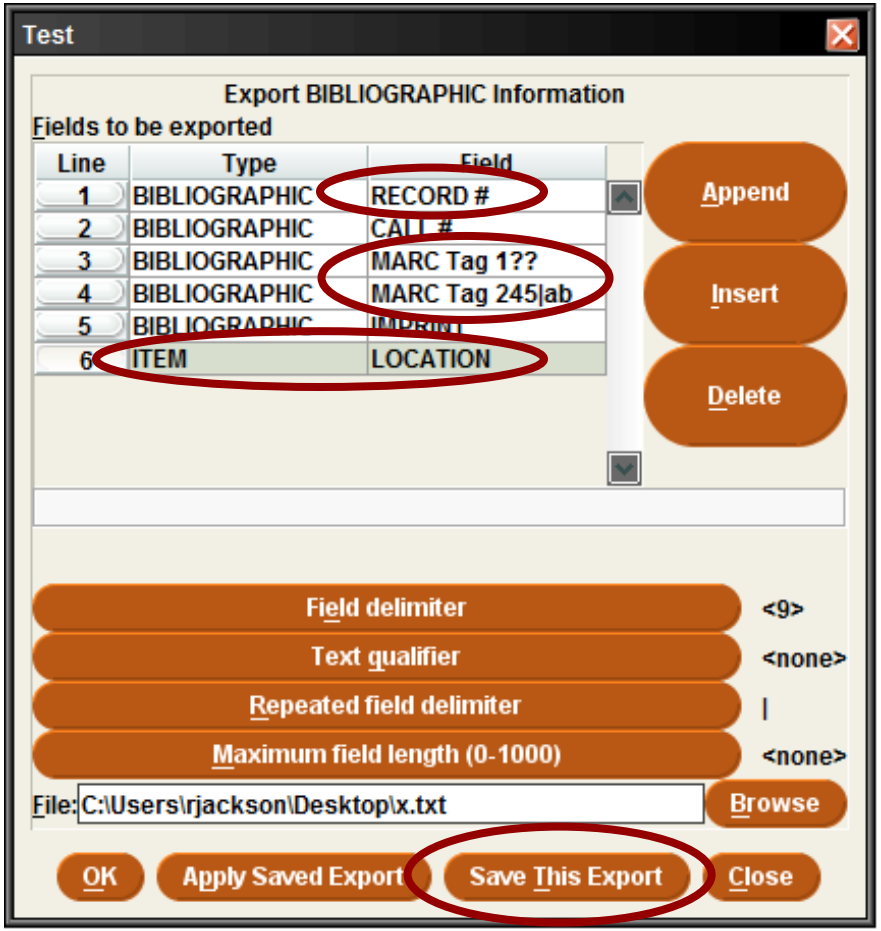


Exporting records

- Sends selected fields to a text file (.txt)
- Fields are delimited (e.g., comma-delimited or tab-delimited), and each record ends with a carriage return
- Sends only the content of the selected fields; you can’t export a full MARC tag
- Commonly used to send data to Excel or a database such as Access

Tips for exporting records

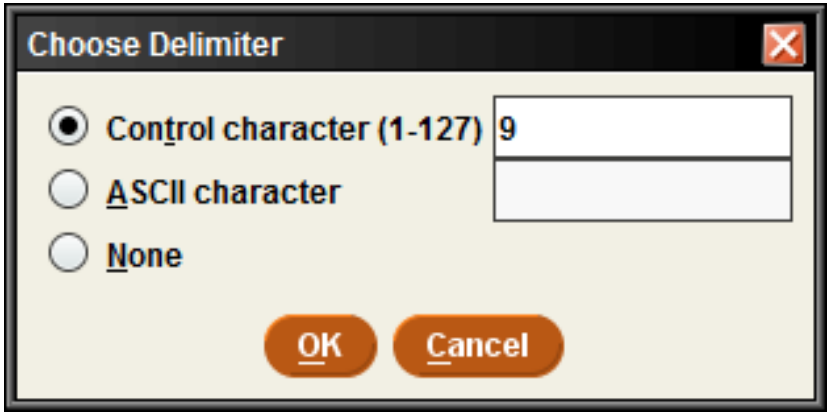
- Include the record #
- Use MARC tags and subfields for more precise data control
- Add data from attached records at the end
- Save the export if you want to use it again



Tips for exporting records

- I prefer to use <Tab> as the field delimiter, rather than the comma
  - No need to put quotes around fields
  - Easily understood by Excel and other programs

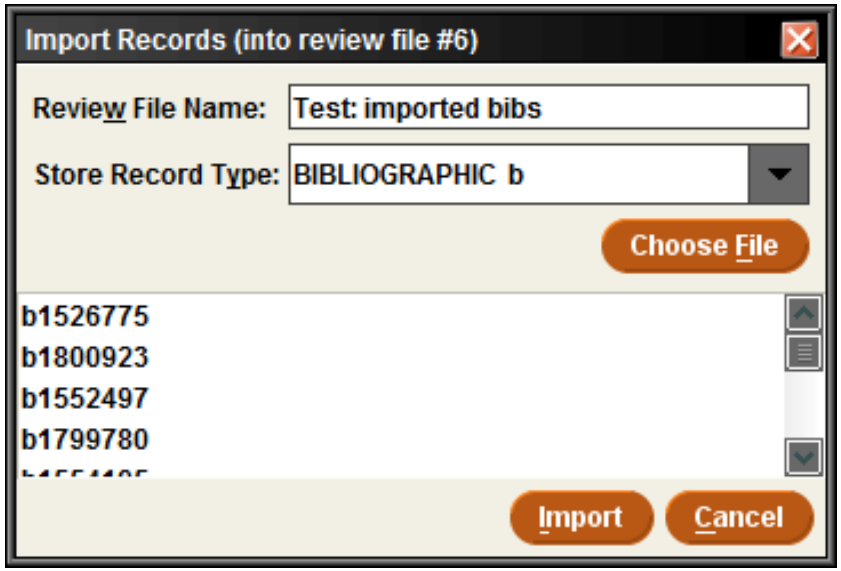
<Tab> is Control character 9:



Importing a review file

Import Records

- With Sierra 2.1+, you can create a review file by importing a file containing a list of record numbers
- File must be plain text
  - Need not have .txt file extension
  - Files in Word or Excel won't work
- Record type must match the type of record numbers found in the file

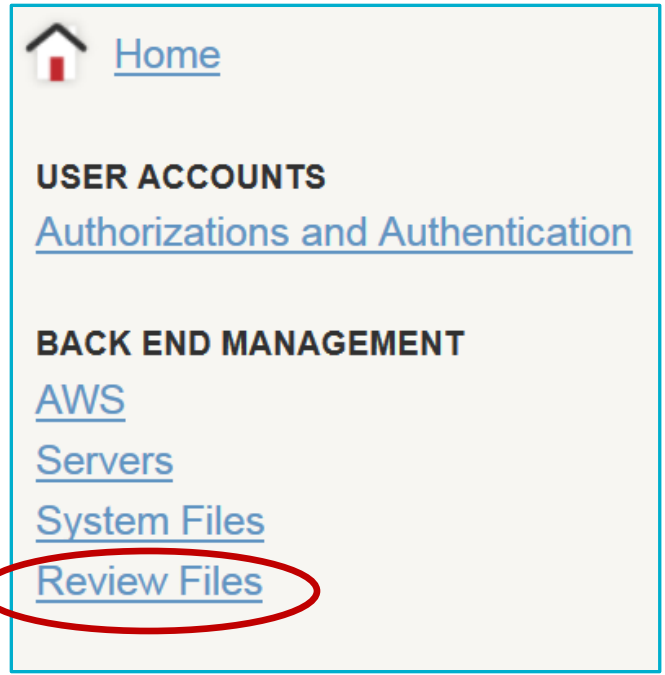


Importing a review file

- The record numbers need not be in any particular format
  - The “dot” is optional
  - The check digit is optional
  - However the letter prefix (b = bib number, i = item number, etc.) is required.
- The record numbers may be mixed with other data — you can even create a review file from a MARC file that contains (for example) the bib record number in a 907 tag

Administering review files (permission 272)

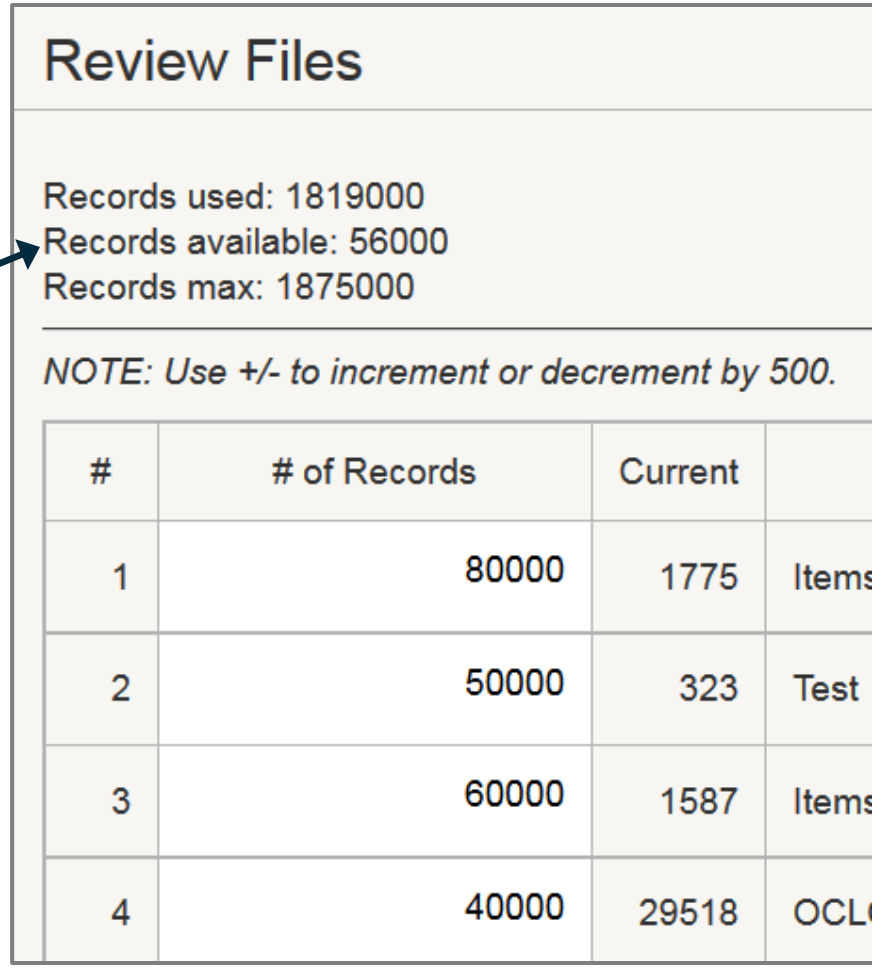
- Each system has a certain number of review file records available (and can acquire more), but they can be allocated among any number of review files
- Under “Review Files” in Sierra Admin
  - Change the size of review files
  - Make additional review files (or remove them)
- Typically, you want a few very large files, and many small or medium size files



Administering review files

Some files have been reduced in size, freeing up 56K records.

These records can be used to increase the size of other review files, or to create several new review files.



Review Files

Records used: 1819000
Records available: 56000
Records max: 1875000

NOTE: Use +/- to increment or decrement by 500.

#	# of Records	Current	
1	80000	1775	Items
2	50000	323	Test
3	60000	1587	Items
4	40000	29518	OCL

Break!

Exercises (Part I)

1) How many bibliographic records cataloged in 2017 (based on Cat Date) either have item records with no barcode, or lack an item record altogether?

Type of record to store: BIBLIOGRAPHIC

Range to search: Full record number range

Search statement(s):

```
      BIB   CAT DATE  between "01-01-2017" and "12-31-2017"
AND ( ITEM  BARCODE   equal to  ""
OR    BIB   LINKED REC  not exist to  ITEM  )
```

2) Find records for items purchased with unrestricted funds (Fund code = unres) that have now been flagged as missing (Status = m).

Type of record to store: BIBLIOGRAPHIC

Range to search: Full record number range

Search statement(s):

```
        ORDER   FUND     equal to  "unres"
  AND   ITEM    STATUS   equal to  "m"
```

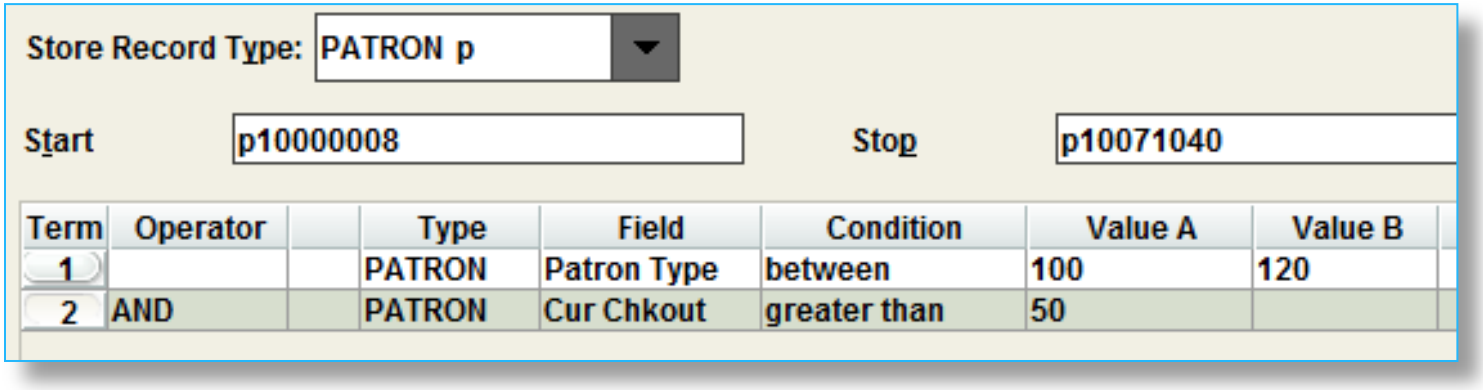
3) What percentage of items with call numbers in LC class “F” have never been checked out? (Hint: 2 review files are needed.)

Type of record to store:	ITEM
Range to search:	Index – LC Call #: ‘F’ to ‘F 9999’
Search statement(s):	*[no search statement needed (unless you need to exclude suppressed records, records w/o Cat Date, Lib Use Only, etc.]*

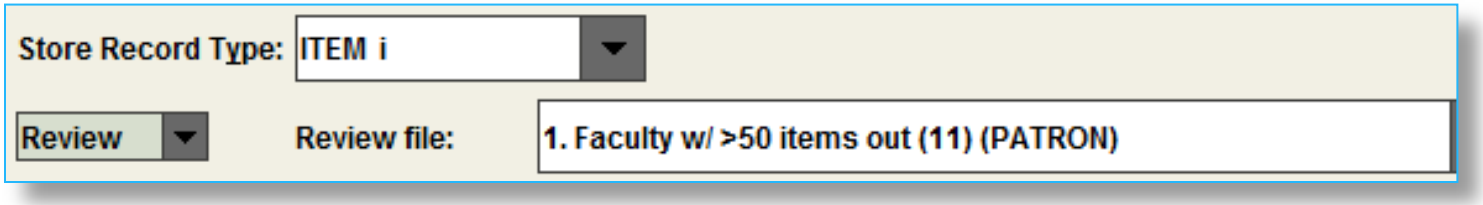
Type of record to store:	ITEM
Range to search:	Review: *[the above review file]*
Search statement(s):	`ITEM TOT CHKOUT equal to 0`

(Divide the second count by the first to obtain the percentage.)

4a) How many faculty members (Patron Types 100-120) have more than 50 items checked out?



4b) Make a list of those items.



(No search statement needed.)

5) *The National union catalog, pre-1956 imprints* (record # .b14649640) has an item record for each of its 754 volumes. Quickly make a review file of those items).

Type of record to store: ITEM

Range to search: Full (or limited) record number range

Search statement(s):

```
BIBLIOGRAPHIC RECORD # equal to ".b14649640"
```

6) Consider this search statement:

```
BIBLIOGRAPHIC  TITLE  has  "500"
```

Which of the following fields will be matched by this search?

☑ 245 14 |aThe 500 hats of Bartholomew Cubbins /|cby Dr. Seuss.

☑ 245 10 |aLife in Sing Sing /|cby Number 1500

☒ 740 2 |a500 years of Italian master drawings …

? 245 00 |aAmerindian signs :|b5,000 years of Precolumbian art …

In Sierra, “has” only sees the content of the field:

✗ `|aAmerindian signs :|b5,000 …`

But in Millennium, “has” looks at the whole MARC tag:

✓ `24500|aAmerindian signs …`

However, in Sierra these searches:

```
BIBLIOGRAPHIC  TITLE  matches  "500"
BIBLIOGRAPHIC  MARC tag 245  has  "500"
```

also match the “500” in “`24500|aAmerindian…`”

4) Using regular expressions

Using regular expressions in Create Lists

- A powerful text processing tool that
- Allows “fuzzier” matching, or finding particular patterns of data rather than specific values
- Widely used in many computer applications
- Invoked in Create Lists with the “matches” operator

A handout is available that provides additional information to this overview.

Literal characters and metacharacters

- Literal characters: normal text characters that represent themselves in the match

  They include:

  A-Z a-z 0-9 <space> | most punctuation

- Metacharacters: perform some function in the regular expression

  They include:

  . [ ] + * { } ? ( ) ^ $ \

Matching any character – the “dot”: .

- Period (or “dot”) matches any single character

Problem: **Limit a search to titles published in the United States**

Solution:

```
COUNTRY matches "..u"
```

Character classes: […]

- Represents any single character that is a member of the user-defined class

Example	***Matches***
`[aeiou]`	any of the letters *a*, *e*, *i*, *o*, or *u*
`['"]`	a single quote or double quote
`[a-z]`	any letter (upper or lower case)
`[a-z0-9]`	any letter or number
`[14-79]`	any of the numbers 1, 4, 5, 6, 7, or 9
`[- ,.]`	a hyphen, space, comma, or period

Character classes: […]

Problem: Limit a search to records where BCODE3 in the bibliographic record is ‘l’, ‘n’, or ‘z’.

Solution:

```
BCODE3 matches "[lnz]"
```

Negated character classes: [^ …]

- Represents any single character that is *NOT* a member of the defined class

Example	***Matches***
`[^ ]`	**any character that is not a space**
`[^0-9]`	**any character that is not a number**
`[^avx-z]`	**any character except *a*, *v*, *x*, *y*, or *z***

Negated character classes: [^ …]

- Negated character classes are particularly useful for finding invalid data

Problem: Some MARC fields have missing (or invalid) subfield codes. For example:

```
                                                y
650  0 |aUnited States|xHistory|1865-1898.
                                                ^
```

Negated character classes: [^ …]

Solution:

1) Determine what subfield codes are valid for the particular MARC tag.

   For MARC tag 650, the valid codes are:
   a, v, x, *y, and* z. *(There could be others.)*

2) Construct a regular expression that matches a subfield delimiter ("|") followed by any character that is *not* one of the valid codes:

```
MARC tag 650 matches "|[^avxyz]"
```

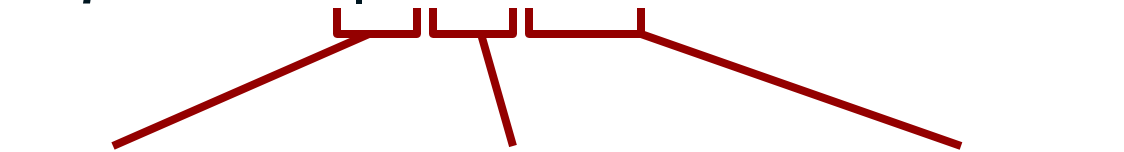
Quantifiers: * + {*min*,*max*} {*num*} ?

- Do not themselves represent any characters
- Apply to the preceding character (or string of characters), allowing some number of occurrences
- Can apply to character classes as well as literal characters

Quantifiers – the asterisk (“star”): *

* Preceding character(s) occur 0 or more times

Example: "|a0*523"



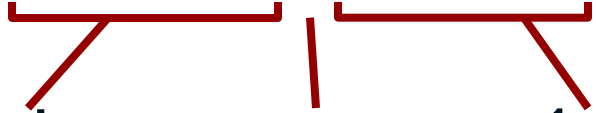
“|a” “0”, any number of times “523”

Matches: |a523
|a0523
|a00523 *[etc.]*

Quantifiers – the plus: +

\+ Preceding character(s) occur 1 or more times

Example: `"[0-9]+ [a-z]+"`



1 or more numbers **space** **1 or more letters**

Matches:

1812 Overture
76 trombones
101 photographs
7 arts

Quantifiers: {*min* , *max*}

{*min* , *max*} Preceding character(s) must occur *min* times, may occur *max* times

Example: `"|a[a-z]{1,3}[0-9]{1,4}"`

Matches:

|aDA670
|aF73
|avid0023

Quantifiers – the question mark: ?


Sierra only!

`?` Preceding character(s) is optional (occurs 0 or 1 times)

Examples: `"colou?r"` `" ?-- ?"`

Matches: **“color” or “colour”**

“--” with or without surrounding spaces

In Millennium, “?” is a literal — use “`{0,1}`” instead

Grouping: (…)

- Allows a quantifier to apply to a string of multiple characters
- Especially useful for making a string of characters optional — by adding “?” (or “{0,1}”)

Example:

```
"|a(oversize )?[a-z]{1,3}[0-9]{1,4}"
```

Matches: **|aND237**
|aOversize F1219

“Dot-star”: .*

- A metasequence that represents any number of unspecified characters (including none)

Problem: **Find subfield codes that are repeated, but should not be, for example:**

```
245 10 |a1876 :|ba novel /|by Gore Vidal.
245 10 |aCalifornia :|ca history /|cAndrew F. Rolle.
```

Solution:

```
MARC Tag 245 matches "|b.*|b"
         OR  matches "|c.*|c"
```

Position indicators – the dollar sign: $

- Represents the end of field *position* (does not itself stand for a character)
- Anchors what precedes it to the end of the field
- Must appear last in the regular expression

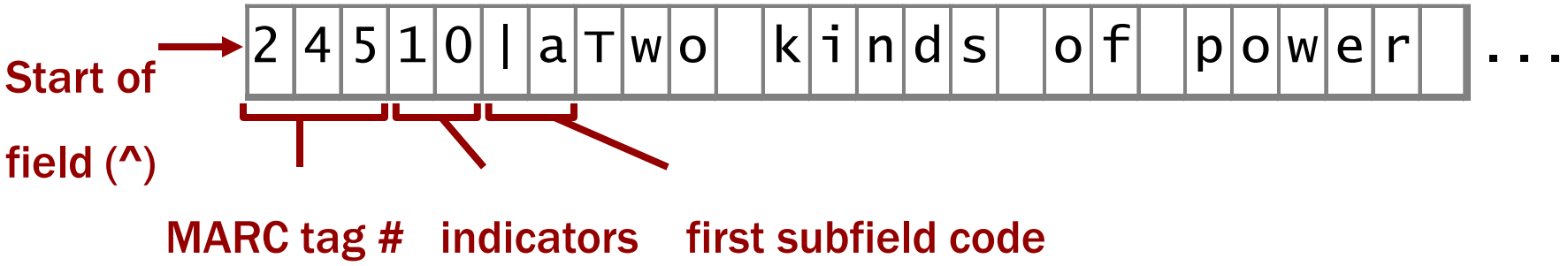
Example: `CALL # matches "196[0-9]$"`

Matches: “090 |aQK47|b.F87 1967”

Does <u>not</u> match: “090 |aPE1963|b.C5”

Position indicators – the circumflex: ^

- Represents the start of field *position* (does not itself stand for a character)
- Anchors what follows it to the start of the field
- Must appear first in the regular expression
- For MARC variable tags, the field begins with the tag number



➢ ***More information on Handout page 3 (Section 2)***

Treating a metacharacter as a literal: \

Millennium only!

- The backslash indicates that the following character should be treated as a literal

Example	***Matches***
`\.\.\.$`	**3 periods at the end of field**
`\$[12][0-9]{2}`	**A dollar sign followed by "1" or "2", followed by 2 more numbers (i.e. $100 - $299)**

This is supposed to work in Sierra!
(It has gone to Software Engineering.)

Treating a metacharacter as a literal: […]

Oddly, Sierra treats "\" as a literal; use character classes instead. (This also works in Millennium.)

Examples	***Matches***
`[.][.][.]$`	**3 periods at the end of field**
`[.]{3}$`	**(alternate version)**
`[$][12][0-9]{2}`	**$100 - $299**

To find fields ending with a question mark:

`?$`	**Works in Millennium only**
`[?]$`	**Works in Millennium & Sierra**

Putting it together …

The real power of regular expressions comes from combining metacharacters in various ways

Examples

Problem: **Limit a search to titles published *outside* of the United States.**

Solution:

```
COUNTRY matches "^..[^u]"
```

Variation (Canada outside of Ontario and Quebec)

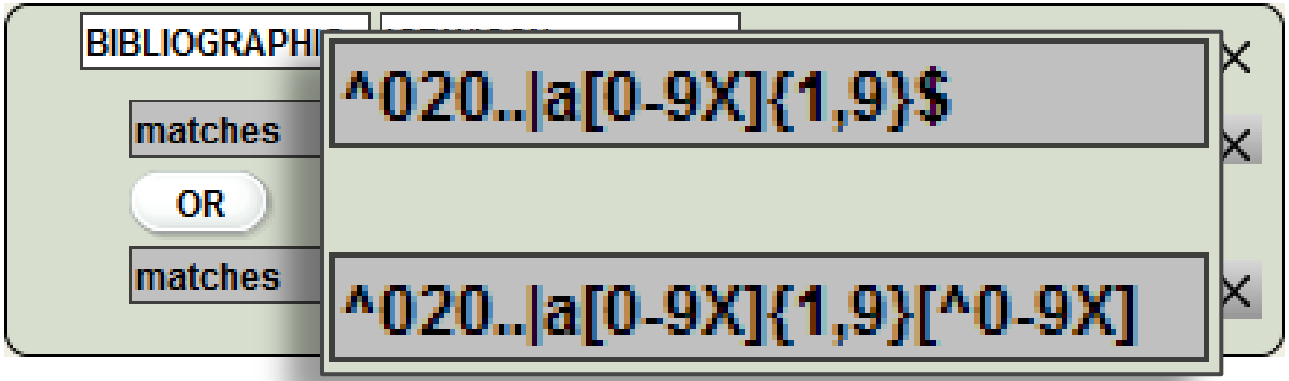
```
COUNTRY matches "[^oq].c"
```

There is an odd quirk in Sierra regarding the Country field. Details in the Handout p. 10.

Examples

Problem: Find ISBNs with fewer than 10 characters.

Solution:



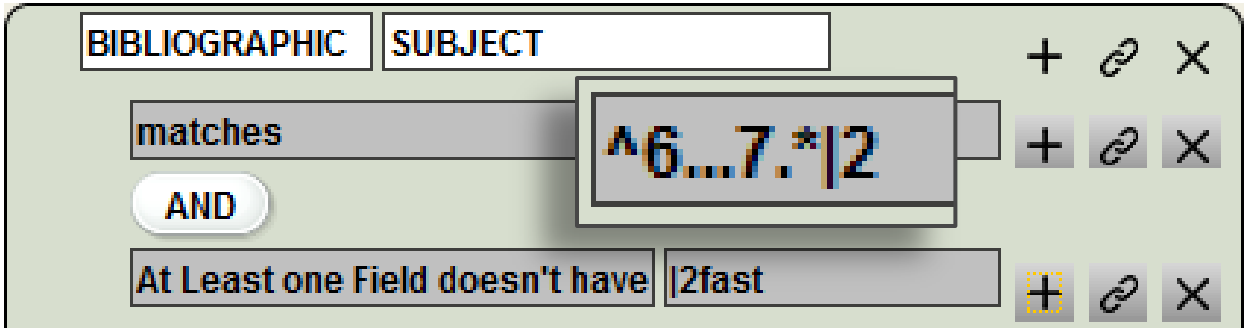
To find a string of characters shorter than a certain length, the string must be "anchored" at both ends. In this case, "|a" marks the beginning and "$" or "[^0-9X]" marks the end.

This will also find ISBNs such as: "0-7134-1474-x". Can you tell why?

Examples

Problem: Find subject headings (6xx) with 2nd indicator 7 in which the subfield 2 code is not “fast”.

Solution:



Matches headings such as: 610 27 |aKenwood Vineyards.|2local

Negative conditions — finding the absence of the string “|2fast” in this case — are difficult to do with a regular expression in Create Lists. Fortunately, the Enhanced query builder enables a simpler solution.

Examples

Problem: **Find title statements in MARC tag 245 that have only a subfield |a and its content.**

Solution:

```
TITLE matches "^245..|a[^|]+$"
```

This doesn't work in Sierra!

What's up with "[^|]" in Sierra?

- The simplest way to match the content of any subfield is with: "[^|]+"
- Works in Millennium, is supposed to work in Sierra (per the manual), but Sierra will fail to find any records
- Workaround: "([-{}~]*[^[:ascii:]]*)+"

POSIX character class

Solution to previous problem — <u>in Sierra</u>:

```
TITLE  matches
   "^245.+|a([ -{}~]*[^[:ascii:]]*)+$"
```

➢ ***Details in the handout, page 6***

Exercises (Part II)

1) Find Patron records with barcodes that do not begin with “2”

```
PATRON  BARCODE  matches  "^[^2]"
```

2) Find Items with barcodes that are longer or shorter than 14 characters.

```
ITEM  BARCODE
   matches    ^.{1,13}$
     OR
   matches    .{15}
```

3) Find records where the 041 has a non-blank second indicator.

```
BIB  MARC [field tag 'y']  matches "^041.[^ ]"
```

4) Find any 035s that contain non-numeric characters.

```
BIB  MARC tag 035  matches  "|a.*[^0-9]"
```

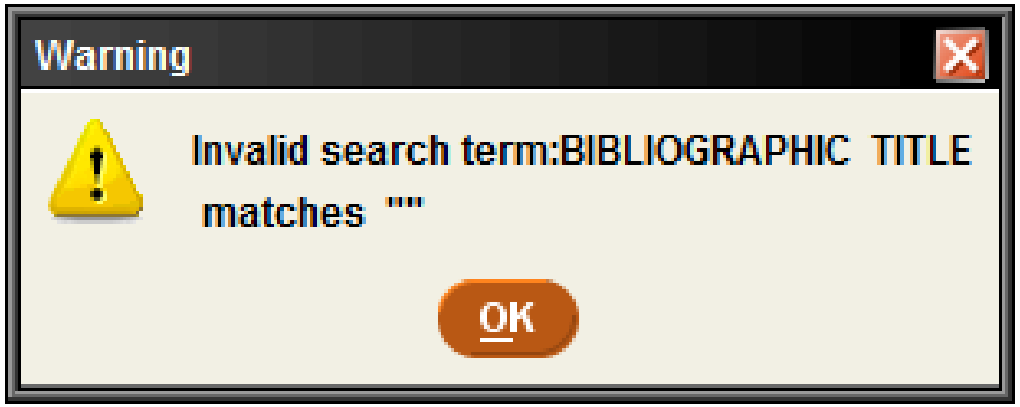
5) Find bibliographic records where the 245 field has a “|c” that is not the last subfield delimiter.

```
BIB  MARC tag 245  matches  "|c.*|"
```

What did I do wrong?

If you see an error message like this when you click Search:

You probably did nothing wrong.



In the Classic query builder, when you use “has” or “matches”, you must tab or click away from the operand (Value A) before clicking Search.

5) JSON and the ‘in’ operator

Record property fields

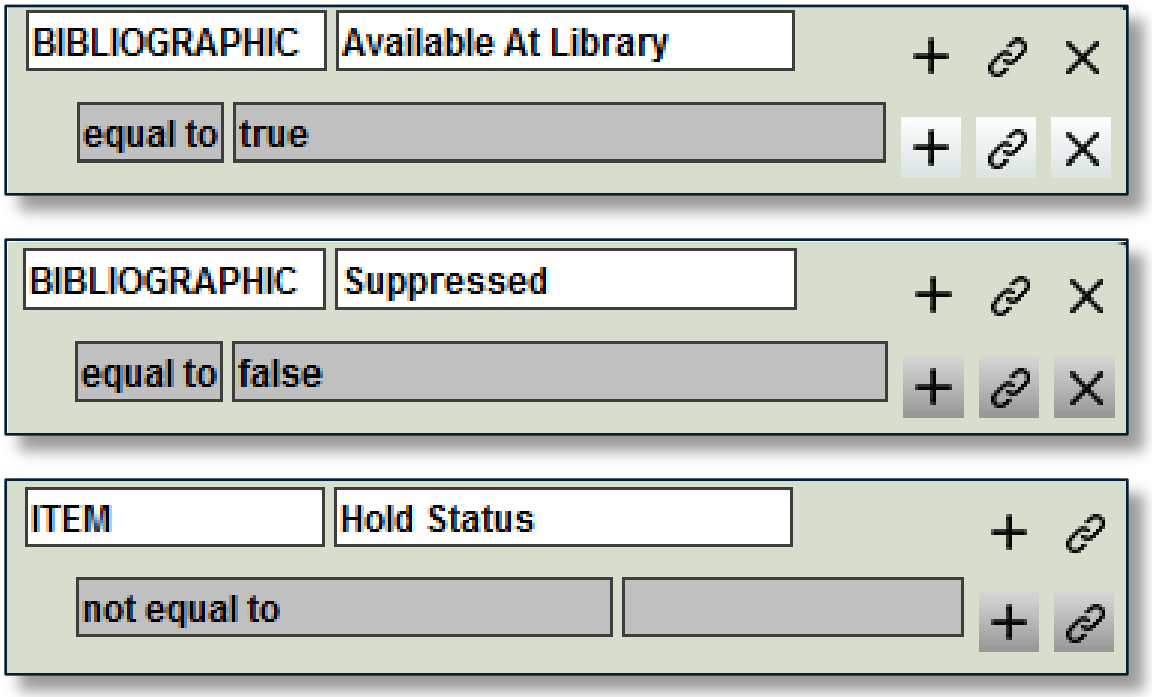
For each record type, Sierra provides several “special fields” dealing with various record properties

- Bib records: Available at Library
- Bib records: Suppressed
- Bib records: Publish year
- Bib records: On Course Reserve
- Item records: Hold Status, Hold Patron, etc.
- Patron records: First, middle, last name; Addr1, Addr2

See: Sierra Guide > Creating Lists (Review Files) > Using Boolean Searching > Using Property Fields in Searches.

Record property fields

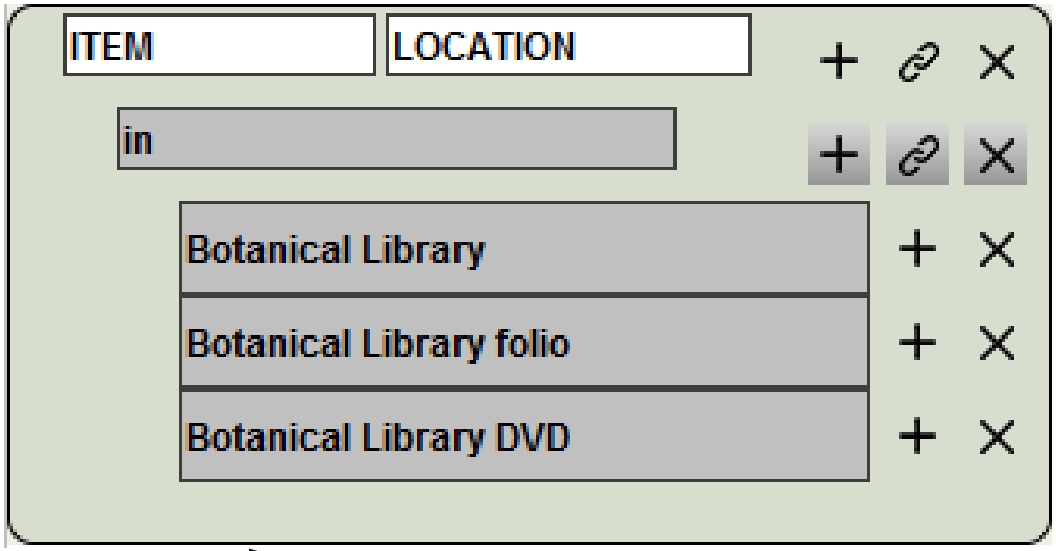
- Some examples:



The JSON query builder

- Create Lists queries created in Classic or Enhanced can also be viewed as JSON
- JSON = JavaScript Object Notation
  - A “lightweight” data-interchange format
  - Easy for humans to read and write, easy for machines to parse
- JSON queries can be:
  - saved to your computer as text files
  - copy/pasted as new queries
  - easily shared with others
  - used in API calls

The JSON query builder



Enhanced

JSON

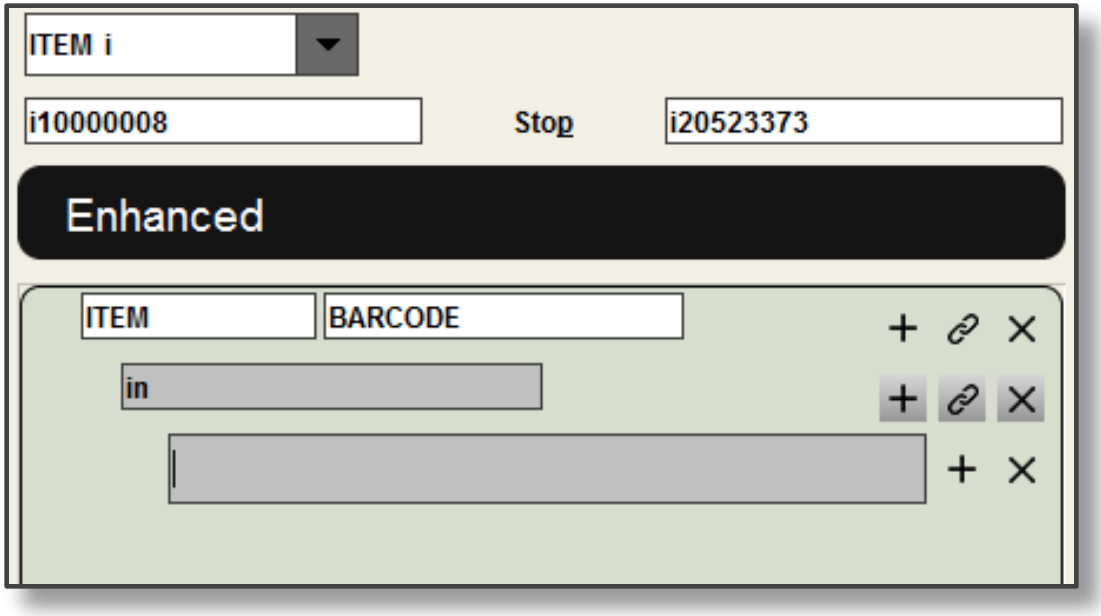
```
{
  "queries": [
    {
      "target": {
        "record": {
          "type": "item"
        },
        "id": 79
      },
      "expr": [
        {
          "op": "in",
          "operands": [
            "bot  ",
            "botf ",
            "bvid "
          ]
        }
      ]
    }
  ]
}
```

Saving JSON queries

Turning a list of barcodes into a review file

- You can take advantage of JSON's plain text format to paste in a list of barcodes as operands to an "IN" operator

- Start in Enhanced view:
  - Store Item records
  - Set the field and operator (Item Barcode in …)



Turning a list of barcodes into a review file

- Switch to JSON:

JSON

```
{
  "queries": [
    {
      "target": {
        "record": {
          "type": "item"
        },
        "field": {
          "tag": "b"
        }
      },
      "expr": [
        {
          "op": "in",
          "operands": [
            ""
          ]
        }
      ]
    }
  ]
}
```

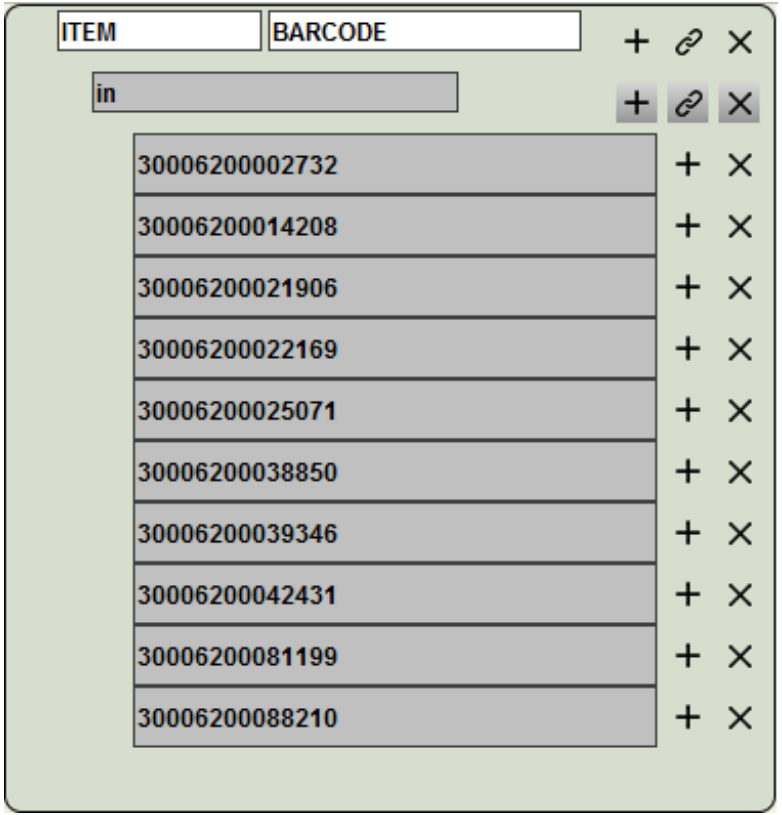
```
"30006200002732",
"30006200014208",
"30006200021906",
"30006200022169",
"30006200025071",
"30006200038850",
"30006200039346",
"30006200042431",
```

Copy/paste the barcode list within the brackets following '"operands":' (barcodes separated by commas)

Turning a list of barcodes into a review file

- Switch back to Enhanced to confirm the query

- Run the search!



Barcodes and the “IN” operator

- Strictly speaking, this is not “converting a list of barcodes to a review file”
- Records will be in record number order, not the order of the barcode list
- If a barcode fails to match a record, you won’t know it (except perhaps by the total number of records retrieved)

The “in” operator functions like “equal to” — any slight difference, even a trailing space, will cause a mismatch.

What about other fields?

- The “in” operator works best with fixed-length fields, or with consistently structured variable-length fields
  - Barcodes (Item and Patron)
  - OCLC numbers
- It *may* work with MARC fields such as call number or system control number (035)
  - If it doesn’t work with a field group tag (c), try using a MARC tag (090)
  - Remember that the field value must *equal* the operand
- Doesn’t work well with ISBNs
- For record numbers, use the Import Records function instead

Further automation of JSON query building

Demonstration of an Excel file (provided with the program materials for this presentation)

- Scan (or paste) barcodes into the spreadsheet
- Copy paste the generated JSON query into the JSON query builder
- Run the search

and more …

Questions?

Richard V. Jackson
Supervising Librarian / System Coordinator
Huntington Library, Art Collections, and Botanical Gardens
San Marino, California
rjackson@huntington.org
http://www.huntington.org/

Barcode scanner

Inateck BCST-10 Wireless Bluetooth Barcode Scanner

- When disconnected from its USB cable, it can store internally up to 2,600 barcodes, downloading them automatically when plugged back in
- It can be configured to Bluetooth with Windows, Android, and iOS devices