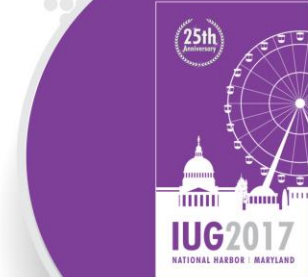


# *Mastering Create Lists*

Richard V. Jackson

The Huntington Library, Art Collections,  
and Botanical Gardens

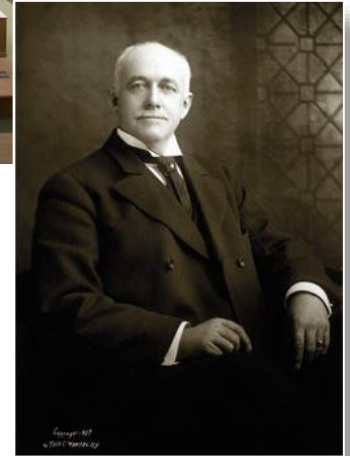
San Marino, California



# 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?

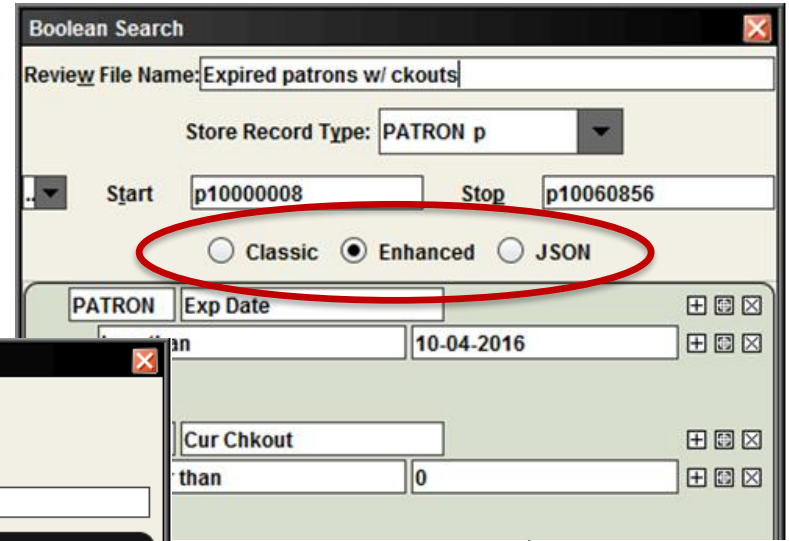
- Create Lists is an essential tool
- Integrated with many functions
- Easier to learn than SQL
- New methods and functionality added in recent Sierra releases

Testing for this presentation done in Sierra 3.0



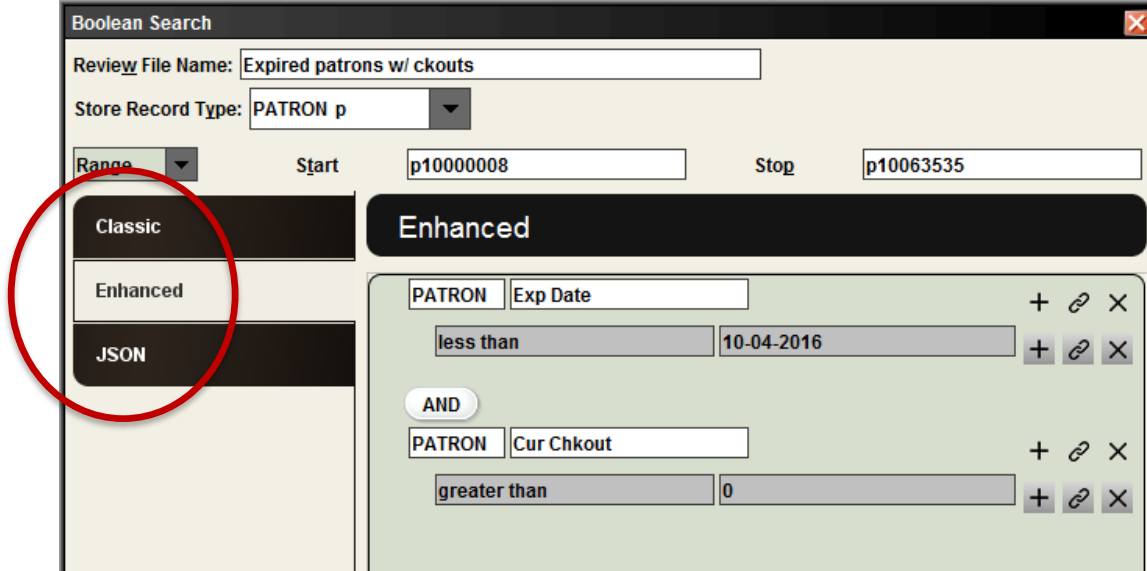
# New query builders

- 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

```
*** INNOPAC -- Copyright 1999, Innovative Interfaces Inc ***
*** MANAGEMENT INFORMATION ***

I > INFORMATION about the system
A > ANALYZE patron searches
G > GATEWAY usage
P > Read PATRON suggestions
R > Read patron REQUESTS

L > Create LISTS of records
S > Create STATISTICAL reports
M > MISC. Acquisitions and Serials

Q > QUIT

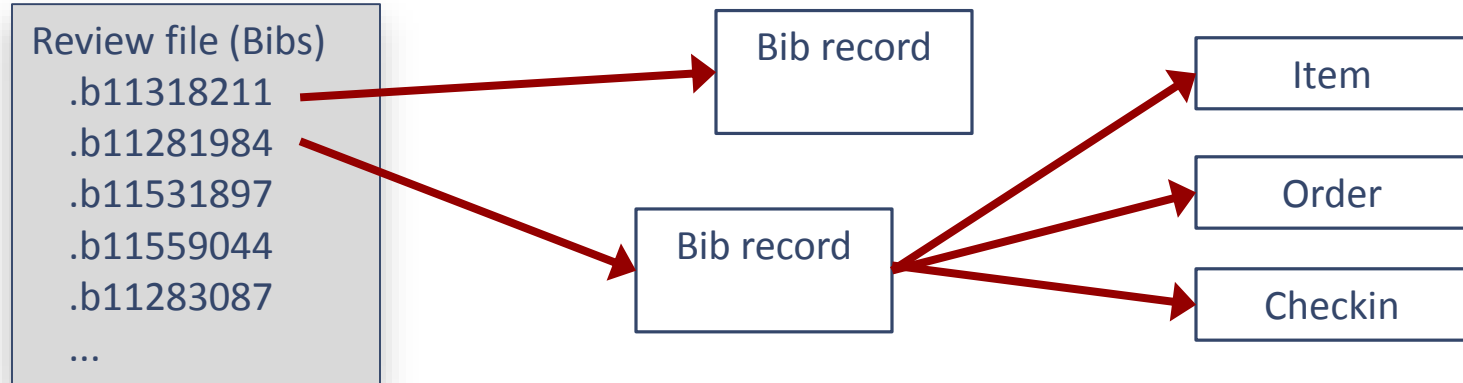
Choose one (I,A,G,P,R,L,S,M,Q) |
```





# 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 vida1.
```

A regular index search:



A screenshot of a search interface. On the left, a dropdown menu shows 'TITLE' selected. To its right, a search box contains the text 'lincoln a novel'. Further right is an orange 'Search' button with a plus sign icon.

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:

W WORD	▼	"jardin botanico"	Search	+
--------	---	-------------------	--------	---

will retrieve a record with this field:

500 "To celebrate the inauguration of the <b>Jardín Botánico</b> 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 geografica  
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** geografica  
italiana

- Result: Searches are case-insensitive



# Changing the base range for searching

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

Range	▼	Start	i2000000a	Stop	i20520190
-------	---	-------	-----------	------	-----------

- You can use “i\*” (or “b\*”, “p\*”, etc.) to indicate the highest record #

- Or search within an index range:

Index	▼	LC CALL NUMBER (c)	▼	Q	to	R
-------	---	--------------------	---	---	----	---

- 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 (assuming the first ‘R’ call number is greater than ‘R’)



# Changing the base range for searching

- Or search within the results of an advanced keyword search:

**Advanced** ▼ **s:oman or s:israel**

- Or search within another review file:

**Review** ▼ **Review file:** **3. Items added 2016 Apr-Jun (excl recats) (1587) (ITEM)** ▼

- 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	
	<b>BIB</b>	<b>CAT DATE</b>	<b>between</b>	<b>01-01-2017</b>	<b>03-31-2017</b>	

Record type  
[target]

Field  
[field identifier]

Condition  
[operator]

Value(s)  
[operands]



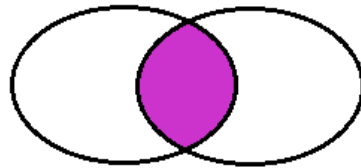
# The “Classic” query builder

Store Record Type:

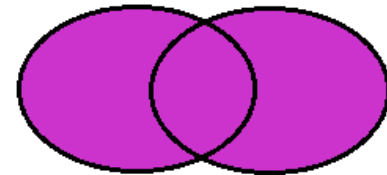
**BIBLIOGRAPHIC b** ▼

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)

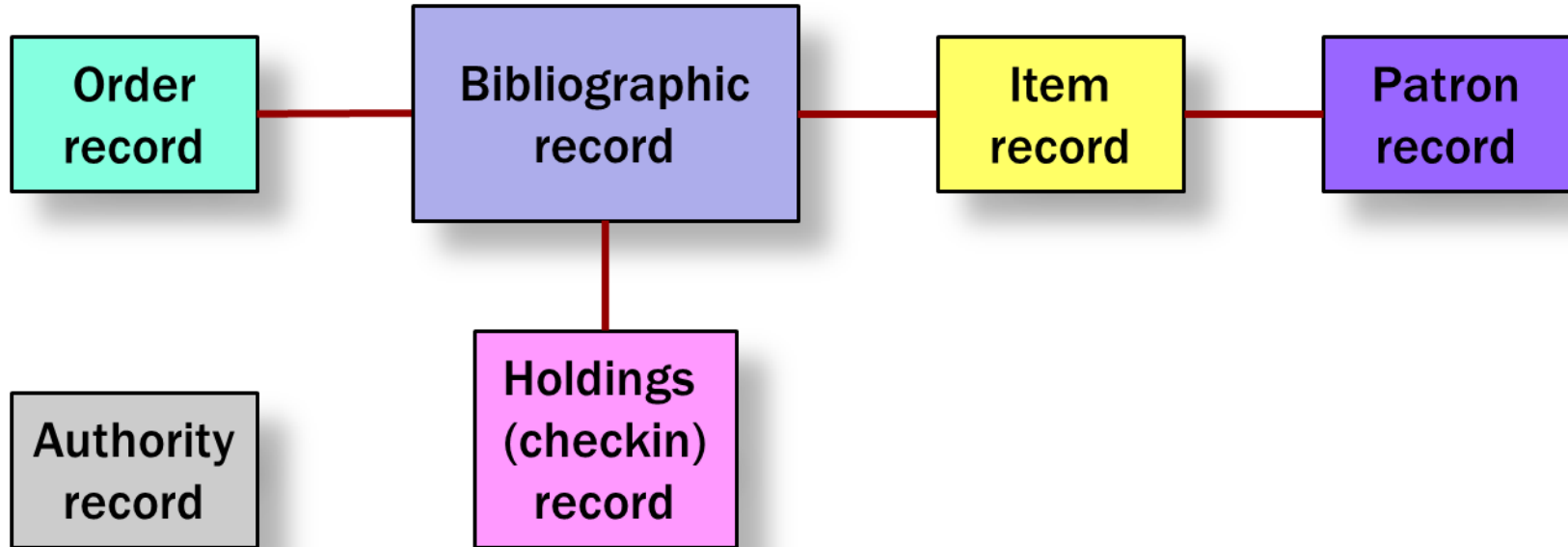


**OR** (either statement  
may be true)



# 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	01-01-2017	03-31-2017
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

French

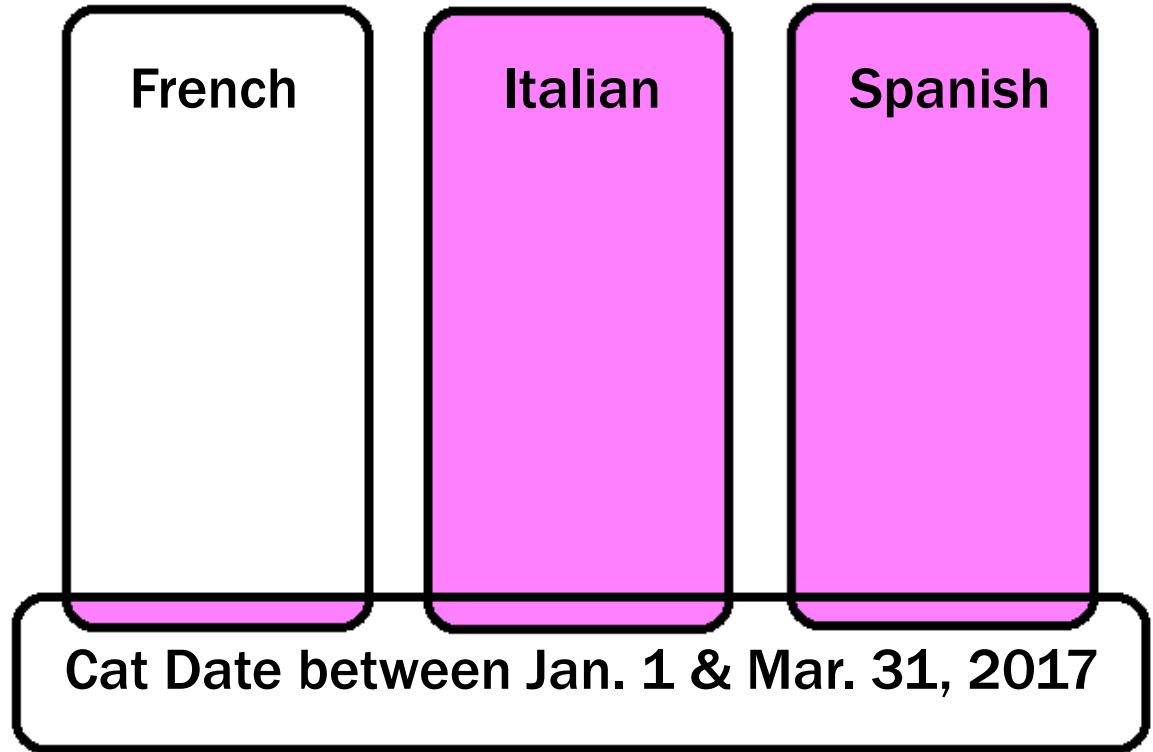
Cat Date between Jan. 1 & Mar. 31, 2017





# 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	01-01-2017	03-31-2017	
AND	(	BIB	LANG	equal to	fre		
OR		BIB	LANG	equal to	ita		
OR		BIB	LANG	equal to	spa		)

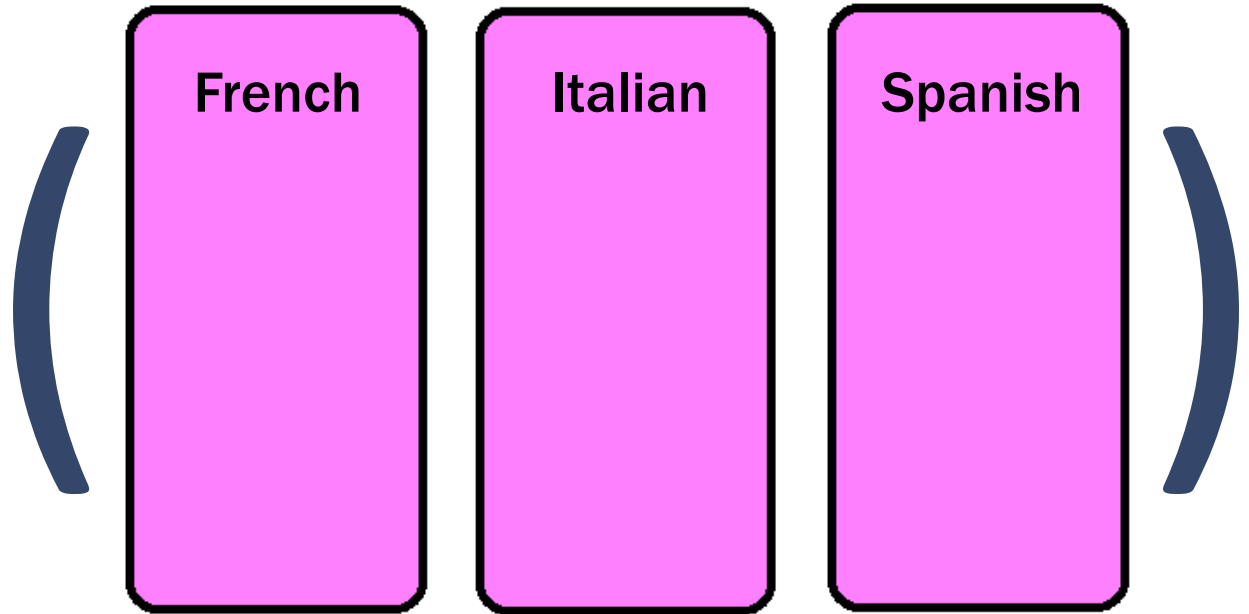
To fix this, group the statements to be evaluated first

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



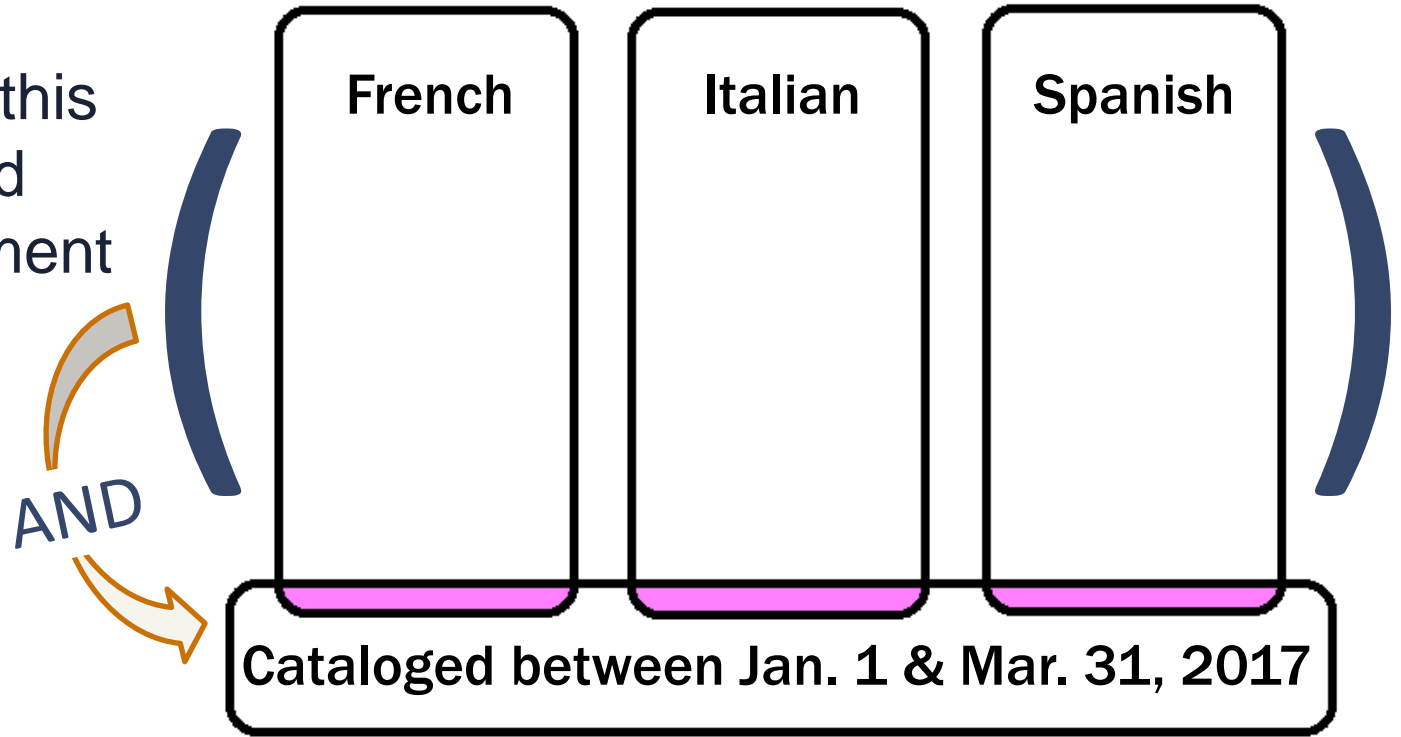
# Complex queries: grouping

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

Type	Field
ITEM	VOLUME

Type	Field
PATRON	Patron Type

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 2<sup>nd</sup> 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
<u>n</u> ot within	n
<u>h</u> as	h
<u>a</u> ll fields don't have	a
at least <u>o</u> ne field	o
doesn't have	
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.

Use 'has' (or 'starts with' or 'ends with') instead

- 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
	<b>BIB</b>	<b>LINKED RECORD</b>	<b>not exist to</b>	<b>ITEM</b>	
<b>AND</b>	<b>BIB</b>	<b>LINKED RECORD</b>	<b>exists to</b>	<b>ORDER</b>	

- 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)





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 equal 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

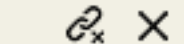
AND

Operand(s)

Boolean operator

BIBLIOGRAPHIC

LANG



equal to

French



OR

BIBLIOGRAPHIC

LANG

equal to

Italian

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

The screenshot shows a query builder interface with two main sections. The top section has a field 'CAT DATE' with a value 'between 01-01-2017 and 03-31-2017'. The bottom section has a field 'LANG' with a value 'equal to French'. Both sections have a logical operator 'AND' and 'OR' respectively. Callouts with red boxes and arrows point to specific icons: 'Remove (delete statement)' points to a red 'X' icon; 'Add condition (new statement)' points to a '+' icon; 'Begin group' points to a chain-link icon; 'Remove grouping' points to a red 'X' icon over a chain-link icon; and 'Add/remove term (operator & operands under the same fieldname)' points to a '+' icon and a chain-link icon.

**Remove (delete statement)**

**Add condition (new statement)**

**Begin group**

**Remove grouping**

**Add/remove term (operator & operands under the same fieldname)**



# 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

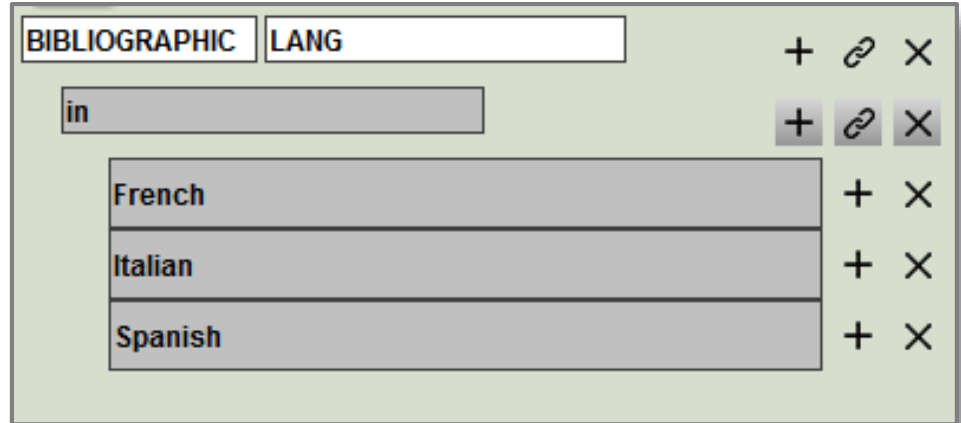
The screenshot shows a search interface with a main query box containing "BIBLIOGRAPHIC MARC Tag 856". Below this, there are two search conditions: "has |z" and "has |3", connected by an "AND" operator. Each condition has a plus sign, a link icon, and a close icon. A red arrow points from the text "one 856 must contain both '|z' and '|3'" to the "has |z" condition. Another red arrow points from the text "This search may find records where '|z' and '|3' occur in different 856s" to the "AND" operator.

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.)



The screenshot shows a search interface with a field labeled 'BIBLIOGRAPHIC' and 'LANG'. The value 'in' is entered in the search box. Below the search box, a list of language options is displayed: French, Italian, and Spanish. Each option has a '+' and an 'x' button next to it, indicating that the search is restricted to these languages.

Field	Value	+	⌋	x
BIBLIOGRAPHIC	LANG	+	⌋	x
in		+	⌋	x
	French	+	x	
	Italian	+	x	
	Spanish	+	x	

## 3) Working with Create Lists



# System-generated review files

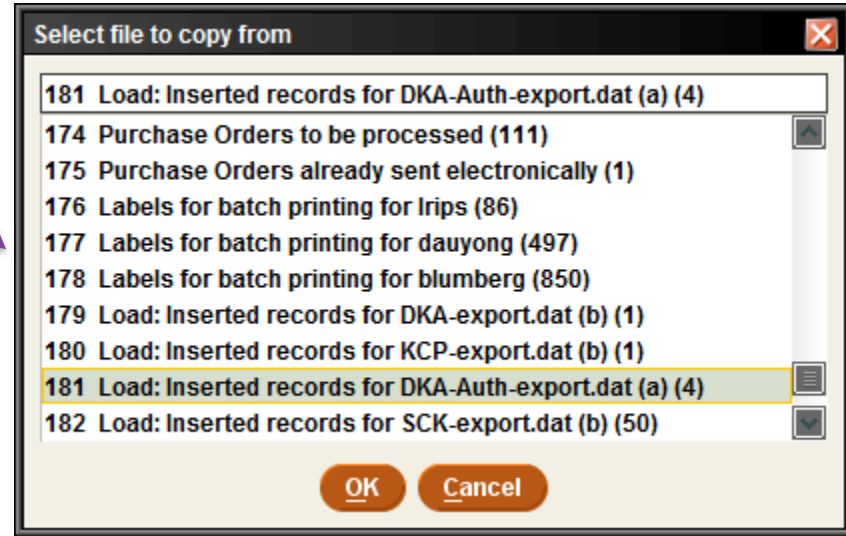
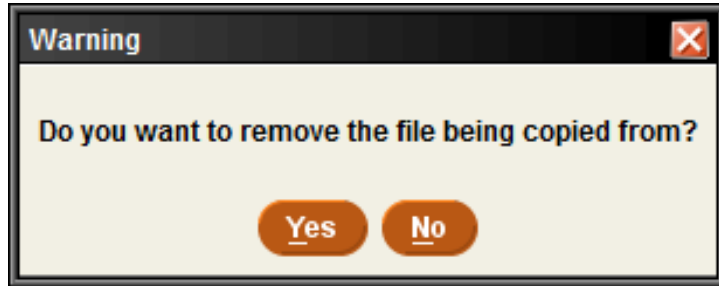
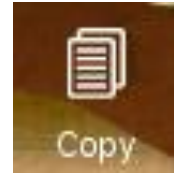
- The system automatically maintains some review files
  - OVERDUE Items
  - Items with HOLDS
  - Items on Holdshelf *[etc.]*
- Other review can be created as part of some process
  - Record loading through Data Exchange  Use Review Files
  - Queue monographic label to print (but not when using print 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."



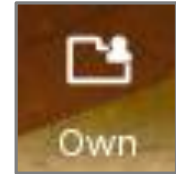
# Owning and releasing review files

- 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 

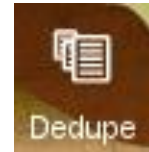
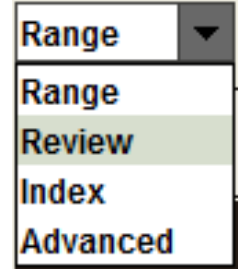
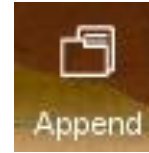
Useful for correcting a mistake in a query without having to reenter the whole thing, or to save a query that had already been executed.





# Merging two review files

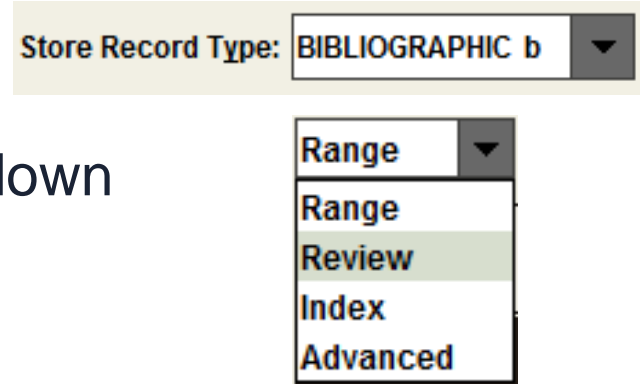
1. Select the first review file and click “Append”
2. Change “Range” to “Review” in the drop-down menu and select the second review file
3. Do not enter any search statements
4. Click “Search”
5. Click “Dedupe” to remove duplicate records
6. If it’s no longer needed, empty the second review file



# 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



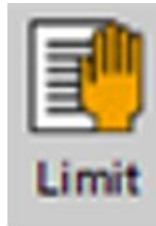
This can be done in reverse (bibs to items), or between any directly linked record types (e.g. items ↔ patrons, orders ↔ bibs, etc.)



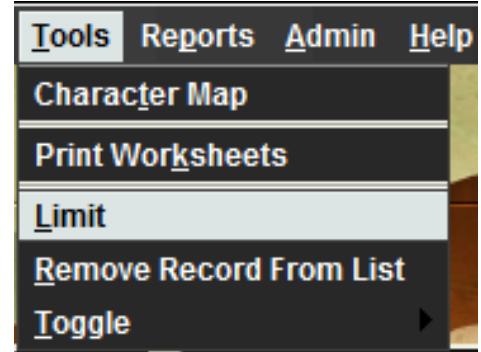
# Using review files in Global Update

- 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.)

RJackson - testing

List ITEM Information

Fields to be listed

Line	Type	Field
1	BIBLIOGRAPHIC	MARC Tag 245 a
2	ITEM	CALL #
3	ITEM	BARCODE
4		

Append

Insert

Delete

Page heading

Starting record (1-14) 1

Ending record (1-14) 14

Number of blank lines between records 1

Number the records in the list

Display meanings of fixed-length fields instead of codes

Display each variable-length field on a new line

Display labels for variable-length fields

If listing bibliographic title, print it in uppercase

Run in background

OK Apply Saved List Save This List Close



# Exporting records

## Export 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

Test

Export BIBLIOGRAPHIC Information

Fields to be exported

Line	Type	Field
1	BIBLIOGRAPHIC	RECORD #
2	BIBLIOGRAPHIC	CALL #
3	BIBLIOGRAPHIC	MARC Tag 1??
4	BIBLIOGRAPHIC	MARC Tag 245 ab
5	BIBLIOGRAPHIC	IMPRINT
6	ITEM	LOCATION

Append

Insert

Delete

Field delimiter <9>

Text qualifier <none>

Repeated field delimiter |

Maximum field length (0-1000) <none>

File: C:\Users\rjackson\Desktop\lx.txt Browse

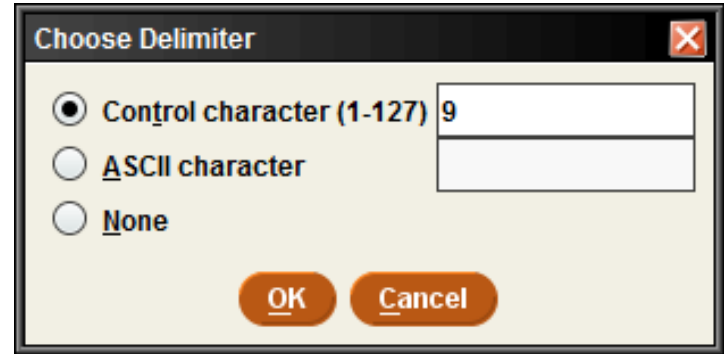
OK Apply Saved Export Save This Export Close



# 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



Import Records (into review file #6)

Review File Name: Test: imported bibs

Store Record Type: BIBLIOGRAPHIC b

Choose File

b1526775  
b1800923  
b1552497  
b1799780  
b1554405

Import Cancel





# Importing a review file

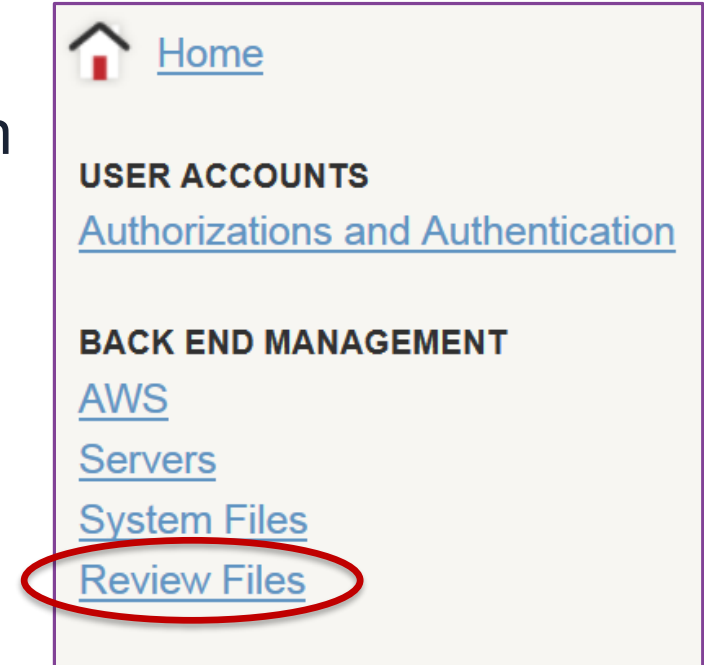
## Import Records

- 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



# Exercises (Part I)

- 1) Find a particular group of holdings records (checkins). You were working on them yesterday, but forgot to add a note with your initials.

Type of record to store: CHECKIN

Range to search: Full record number range

Search statement(s):

CHECKIN UPDATED equal to "04-01-2017"

AND CHECKIN NOTE All fields don't have "rvj"

First statement could also be:

CHECKIN UPDATED equals yesterday



2) Find records with Item location 'hcd' or 'hvid' that were received in 2014 (based on RDate in the Order record).

Type of record to store: **BIBLIOGRAPHIC**

Range to search: **Full record number range**

Search statement(s):

```
( ITEM LOCATION equal to "hcd "
OR  ITEM LOCATION equal to "hvid " )
AND ORDER RDATE between "01-01-2014" & "12-31-2014"
```



- 2) Find records with Item location 'hcd' or 'hvid' that were received in 2014 (based on RDate in the Order record).

*Alternate method using the Enhanced query builder:*

The screenshot shows a query builder interface with the following structure:

- ITEM** (field) **LOCATION** (field) [ + ] [ link ] [ X ]
- equal to** (operator) **Compact discs (request at Reader Servs)** (value) [ + ] [ link ] [ X ]
- OR** (connector)
- equal to** (operator) **Video/DVD (request at Reader Services)** (value) [ + ] [ link ] [ X ]
- AND** (connector)
- ORDER** (field) **RDATE** (field) [ + ] [ link ] [ X ]
- between** (operator) **01-01-2014** (value) [ + ] [ link ] [ X ]
- and** (operator) **12-31-2014** (value)

A red oval highlights the first two conditions: ITEM LOCATION equal to Compact discs (request at Reader Servs) OR equal to Video/DVD (request at Reader Services).



- 3) All bib records from a certain collection have a local note (590 tag) "Mount Wilson Observatory Collection." Find those that have not yet been assigned a call number.

Type of record to store: **BIBLIOGRAPHIC**

Range to search: **Advanced – using:**  
**"mount wilson observatory collection"**

Search statement(s):

**BIBLIOGRAPHIC CALL # equal to ""**



4a) How many items are checked out by students with undeclared majors (PCode3=1)?

The screenshot shows a search criteria form. At the top, a dropdown menu is set to "Store Record Type: ITEM i". Below this, there are fields for "Range", "Start" (i10000008), and "Stop" (i43640503). A table below contains search criteria:

Term	Operator	Type	Field	Condition	Value A	Value B
1		PATRON	PCODE3	equal to	1	

4b) Who are those patrons?

The screenshot shows a search results form. At the top, a dropdown menu is set to "Store Record Type: PATRON p". Below this, there is a "Review" dropdown and a "Review file:" field containing "RJackson - testing (14) (ITEM)". A table below shows the search criteria:

Term	Operator	Type	Field	Condition	Value A	Value B
1						





4b) Who are those patrons?

*Alternate method:*

Store Record Type: PATRON p

Range Start p10000008 Stop p11178796

Term	Operator	Type	Field	Condition	Value A	Value B
1		PATRON	PCODE3	equal to	1	
2	AND	PATRON	CUR CHKOUT	greater than	0	



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 Precolombian art ...



6) Consider this search statement:

BIBLIOGRAPHIC TITLE has "500"

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

- 245 00 |aAmerindian signs :|b5,000 years of PreColumbian art ... |cby
- 245 00 |aAmerindian signs :|b5,000 years of PreColumbian art ...
- 740 01 |aAmerindian signs :|b5,000 years of PreColumbian art ...
- 245 00 |aAmerindian signs :|b5,000 years of PreColumbian art ...



## 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

[aeiou]

['"]

[a-z]

[a-z0-9]

[14-79]

[- , .]

## Matches

any of the letters *a*, *e*, *i*, *o*, or *u*

a single quote or double quote

any letter (upper or lower case)

any letter or number

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: $[^{\dots}]$

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

### Example

$[^ ]$

$[^0-9]$

$[^avx-z]$

### Matches

any character that is not a space

any character that is not a number

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:

651 0 |aUnited States|xHistory|1865-1898.

y  
^



# Negated character classes: [^...]

## *Solution:*

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

*For MARC tag 651, 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 651 matches "| [^avxyz]"



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

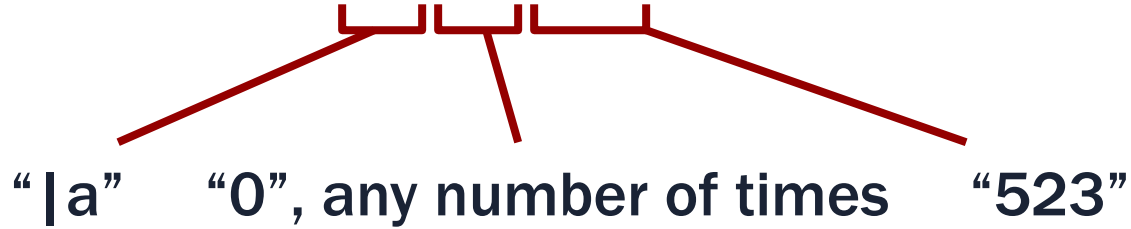
- Do not themselves represent any characters
- Apply to the preceding character (or string of characters), allowing some number of occurrences
- Can apply to character classes as well as literal characters



# Quantifiers – the asterisk (“star”): \*

- \* Preceding character(s) occur 0 or more times

*Example:* " | a0\*523"



*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}"`

“|a”      1-3 letters      1-4 numbers

**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  
|a0versize 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 vida1.

245 10 |aCalifornia :|ca history /|cAndrew F. Rol1e.

**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]\$"

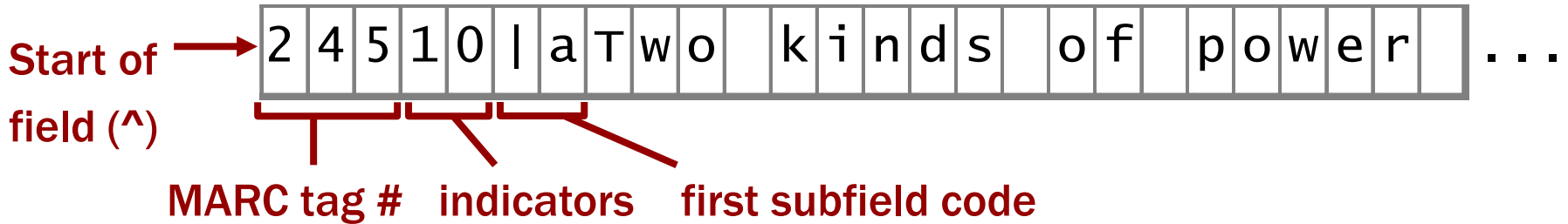
**Matches:** "090 |aQK47|b.F87 1967"

**Does not 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

`\. \. \. $`

`\$ [12] [0-9] {2}`

## Matches

**3 periods at the end of field**

**A dollar sign followed by “1” or “2”, followed by 2 more numbers (i.e. \$100 - \$299)**



# Treating a metacharacter as a literal: [ ... ]

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

## Examples

[.][.][.]\$

[.]{3}\$

[\$][12][0-9]{2}

## Matches

3 periods at the end of field

(alternate version)

\$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 " $\wedge . . [\wedge u]$ "

***Variation (Canada outside of Ontario and Quebec)***

COUNTRY matches " $[\wedge oq] . c$ "

***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:**

The screenshot shows a search interface with a text input field containing "BIBLIOGRAPHI". Below the input field are two search results, each with a "matches" button. The first result shows the regex pattern `^020..|a[0-9X]{1,9}$`. The second result shows the regex pattern `^020..|a[0-9X]{1,9}[^0-9X]`. The interface also includes an "OR" button between the two results.

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 2<sup>nd</sup> indicator 7 in which the subfield 2 code is not “fast”.

**Solution:**

The screenshot shows the Enhanced query builder interface. It features a search bar with the text "BIBLIOGRAPHIC SUBJECT" and a search icon. Below the search bar, there are three query components:

- A "matches" field containing the regular expression `^6...7.*|2`.
- An "AND" button.
- An "At Least one Field doesn't have" field containing the string `|2fast`.

Each component has a plus sign, a link icon, and an X icon for editing or deleting the component.

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:]]\*)<sup>←</sup>+”

POSIX  
character class

*Solution to previous problem — in Sierra:*

TITLE matches

➤ *Details in the handout, page 6*

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



## Exercises (Part II)

- 1) How would you limit a search to records where the call number is in MARC tag 090?

BIBLIOGRAPHIC CALL # matches "^090"

If call numbers are in the Item record:

ITEM CALL # matches "^090"



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

PATRON BARCODE matches " $^{\wedge}[\wedge 2]$ "

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

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





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

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

- 5) Find personal name headings (MARC tag 100, 600, 700) with 19th century birthdates and no death date, e.g:

100 1 worden, J. Perry, |d1866-

(Solution on next slide)



BIBLIOGRAPHIC

MARC Tag

100



matches

|d18[0-9]{2}-\$



OR

matches

|d18[0-9]{2}-[^12]



OR

BIBLIOGRAPHIC

MARC Tag

600



matches

|d18[0-9]{2}-\$



OR

matches

|d18[0-9]{2}-[^12]



OR

BIBLIOGRAPHIC

MARC Tag

100



matches

|d18[0-9]{2}-\$



OR

matches

|d18[0-9]{2}-[^12]



OR

BIBLIOGRAPHIC

MARC Tag

600



This search could be improved; it will match headings such as:

100 1 Lane, Lunsford, |d1803-approximately 1863.

and fail to match headings such as:

100 1 Híjar, Carlos N., |dapproximately 1826-

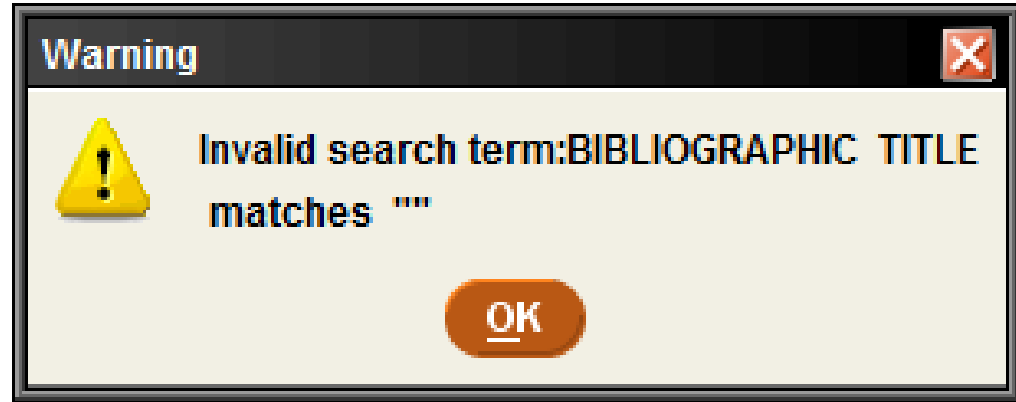


OR

# 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:

BIBLIOGRAPHIC	Available At Library	+		X
equal to	true	+		X

BIBLIOGRAPHIC	Suppressed	+		X
equal to	false	+		X

ITEM	Hold Status	+		
not equal to		+		



# 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 (‘easy’ being relative)
- JSON queries can be:
  - saved to your computer as text files
  - copy/pasted as new queries
  - easily shared with others





# The JSON query builder

ITEM LOCATION + [link] X

in + [link] X

- Botanical Library + X
- Botanical Library folio + X
- Botanical Library DVD + X

Enhanced

JSON

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



# Saving JSON queries

The screenshot shows the 'Boolean Search' window with the following fields and controls:

- Review File Name: IUG 2017 - demo
- Store Record Type: ITEM i
- Range: [dropdown]
- Start: i10000008
- Stop: i20523373
- Enhanced view tabs: Enhanced, JSON
- JSON view content:

```
{
  "queries": [
    {
      "target": {
        "record": {
          "type": "item"
        },
        "id": 79
      },
      "expr": [
        {
          "op": "in",
```
- Buttons at the bottom: Search, Use Existing Search, Retrieve Saved Query, Save JSON, Load JSON, Close

Two callout boxes provide additional information:

- In JSON view, queries can be saved as text files to your workstation** (with an arrow pointing to the 'Save JSON' button)
- Load previously saved JSON queries** (with an arrow pointing to the 'Load JSON' button)



# 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 ...)

The screenshot shows a software interface for creating a list. At the top, there is a dropdown menu labeled "ITEM i". Below it, there are two input fields for barcodes: "i10000008" and "i20523373", with a "Stop" button between them. A large black button labeled "Enhanced" is visible. Below this, there is a list of fields and operators. The first row shows "ITEM" and "BARCODE" with a plus sign, a link icon, and a close icon. The second row shows "in" with a plus sign, a link icon, and a close icon. The third row shows an empty input field with a plus sign and a close icon.



# 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",
"30006200081199"
```

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!

ITEM	BARCODE		
		+	⌂
in		+	⌂
	30006200002732	+	×
	30006200014208	+	×
	30006200021906	+	×
	30006200022169	+	×
	30006200025071	+	×
	30006200038850	+	×
	30006200039346	+	×
	30006200042431	+	×
	30006200081199	+	×
	30006200088210	+	×



# 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 ...





# Barcode scanner

At the end of the presentation, I used this 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



# Questions?

Richard V. Jackson

Supervising Librarian / System Coordinator

Huntington Library, Art Collections, and Botanical Gardens

San Marino, California

[rjackson@huntington.org](mailto:rjackson@huntington.org)

<http://www.huntington.org/>

