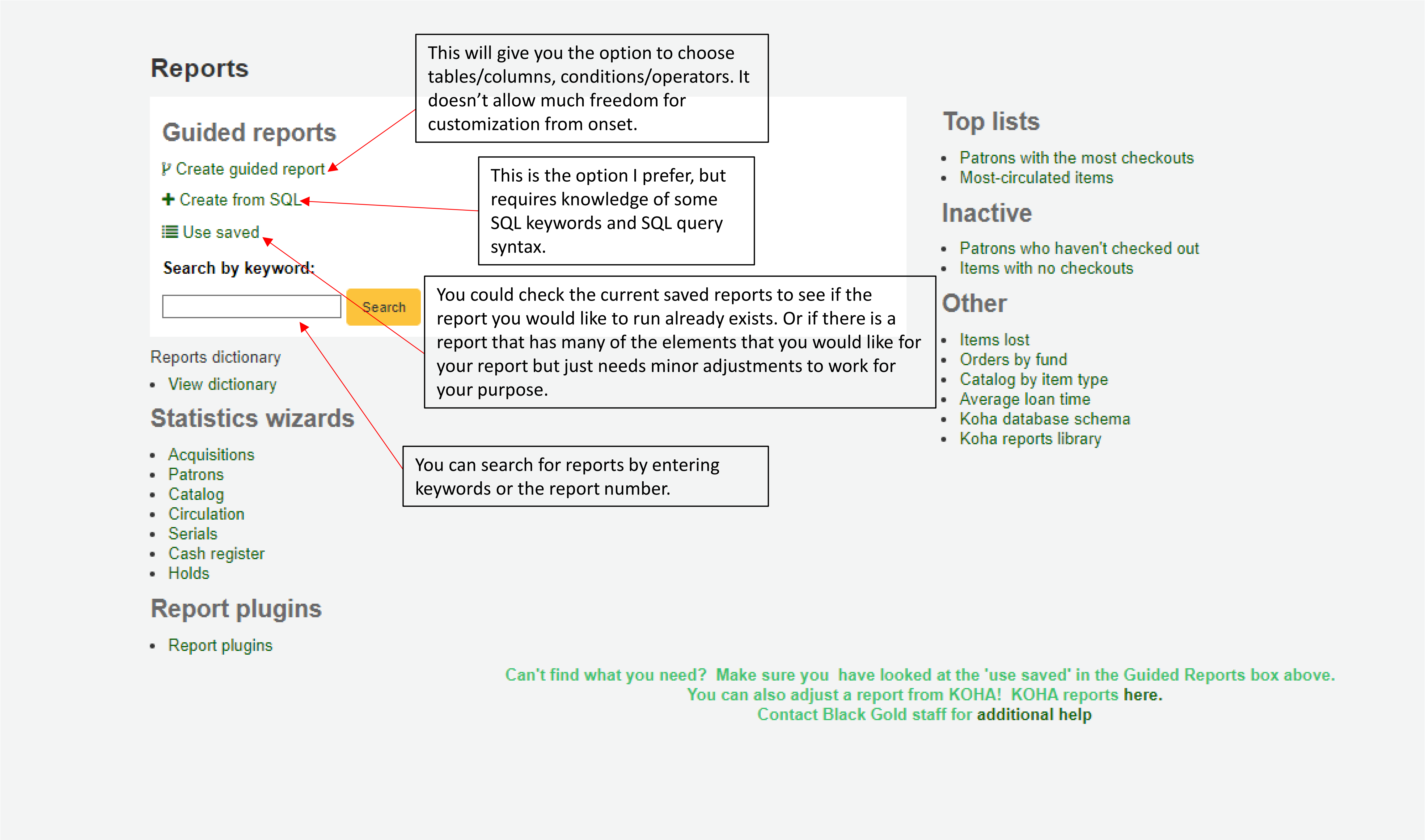
Koha Reports Workshop 9/4/2024

Theo Farias,
Library Technician I
t_farias@ci.lompoc.ca.us
805-875-8790



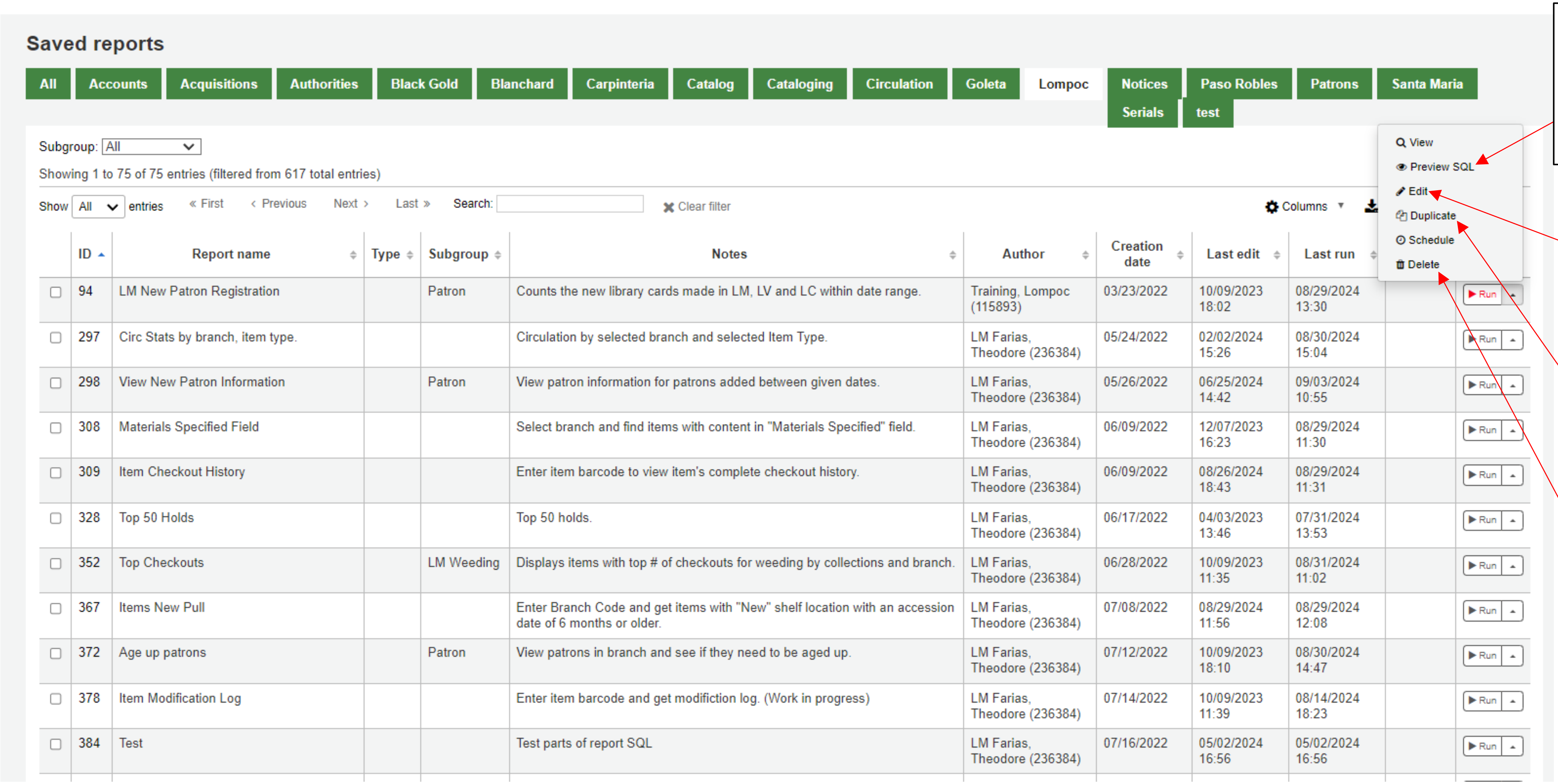
This will give you the option to choose tables/columns, conditions/operators. It doesn't allow much freedom for customization from onset.

This is the option I prefer, but requires knowledge of some SQL keywords and SQL query syntax.

You could check the current saved reports to see if the report you would like to run already exists. Or if there is a report that has many of the elements that you would like for your report but just needs minor adjustments to work for your purpose.

You can search for reports by entering keywords or the report number.

Can't find what you need? Make sure you have looked at the 'use saved' in the Guided Reports box above.
You can also adjust a report from KOHA! KOHA reports here.
Contact Black Gold staff for additional help



Subgroup: All

Showing 1 to 75 of 75 entries (filtered from 617 total entries)

Show All entries

Search:

	ID	Report name	Type	Subgroup	Notes	Author	Creation date	Last edit	Last run
☐	94	LM New Patron Registration		Patron	Counts the new library cards made in LM, LV and LC within date range.	Training, Lompoc (115893)	03/23/2022	10/09/2023 18:02	08/29/2024 13:30
☐	297	Circ Stats by branch, item type.			Circulation by selected branch and selected Item Type.	LM Farias, Theodore (236384)	05/24/2022	02/02/2024 15:26	08/30/2024 15:04
☐	298	View New Patron Information		Patron	View patron information for patrons added between given dates.	LM Farias, Theodore (236384)	05/26/2022	06/25/2024 14:42	09/03/2024 10:55
☐	308	Materials Specified Field			Select branch and find items with content in "Materials Specified" field.	LM Farias, Theodore (236384)	06/09/2022	12/07/2023 16:23	08/29/2024 11:30
☐	309	Item Checkout History			Enter item barcode to view item's complete checkout history.	LM Farias, Theodore (236384)	06/09/2022	08/26/2024 18:43	08/29/2024 11:31
☐	328	Top 50 Holds			Top 50 holds.	LM Farias, Theodore (236384)	06/17/2022	04/03/2023 13:46	07/31/2024 13:53
☐	352	Top Checkouts		LM Weeding	Displays items with top # of checkouts for weeding by collections and branch.	LM Farias, Theodore (236384)	06/28/2022	10/09/2023 11:35	08/31/2024 11:02
☐	367	Items New Pull			Enter Branch Code and get items with "New" shelf location with an accession date of 6 months or older.	LM Farias, Theodore (236384)	07/08/2022	08/29/2024 11:56	08/29/2024 12:08
☐	372	Age up patrons		Patron	View patrons in branch and see if they need to be aged up.	LM Farias, Theodore (236384)	07/12/2022	10/09/2023 18:10	08/30/2024 14:47
☐	378	Item Modification Log			Enter item barcode and get modifiction log. (Work in progress)	LM Farias, Theodore (236384)	07/14/2022	10/09/2023 11:39	08/14/2024 18:23
☐	384	Test			Test parts of report SQL	LM Farias, Theodore (236384)	07/16/2022	05/02/2024 16:56	05/02/2024 16:56

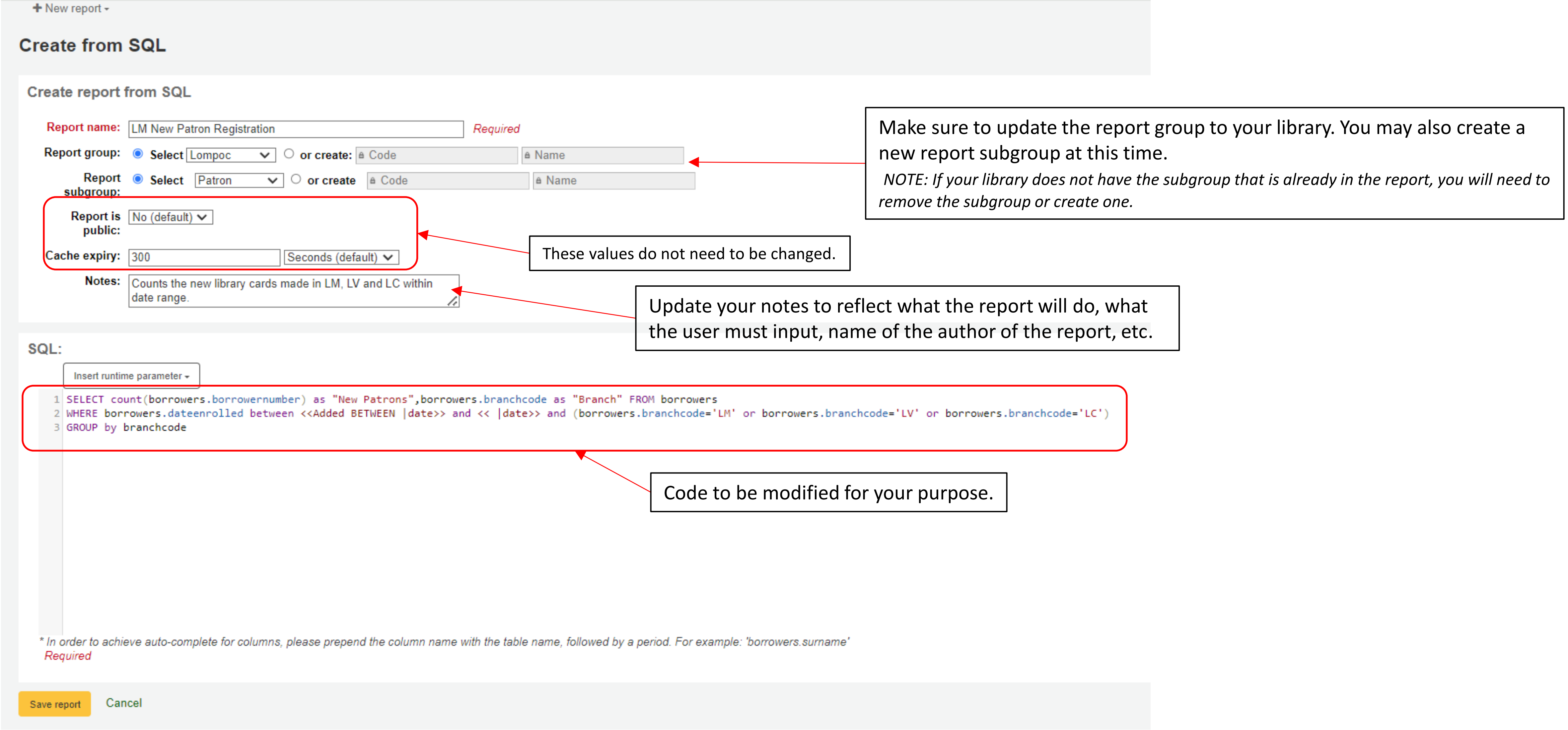
Click on “Preview SQL” to see the SQL code of the report. This is useful for copying parts of code to use in other reports.

Never edit other libraries’ reports! You may edit your own reports by clicking on “Edit”.

Click on “Duplicate” to copy the report and it can be added to your report library.

Never delete other libraries’ reports! You may delete your own reports by clicking on “Delete”.

- When looking for a report to duplicate and adjust, you could select one of the tabs at the top to narrow down what type of report you are looking for (Accounts, Acquisitions, etc.).
- If you know what library the report was created by, you can select on their tab and click on the arrow next to run to bring up a menu where you can choose what you would like to do with the report.



Create from SQL

Create report from SQL

Report name: LM New Patron Registration *Required*

Report group: ● Select Lompoc ○ or create: Code Name

Report subgroup: ● Select Patron ○ or create Code Name

Report is public: No (default)

Cache expiry: 300 Seconds (default)

Notes: Counts the new library cards made in LM, LV and LC within date range.

SQL:

```
SELECT count(borrowers.borrowernumber) as "New Patrons",borrowers.branchcode as "Branch" FROM borrowers
WHERE borrowers.dateenrolled between <<Added BETWEEN |date>> and << |date>> and (borrowers.branchcode='LM' or borrowers.branchcode='LV' or borrowers.branchcode='LC')
GROUP by branchcode
```

** In order to achieve auto-complete for columns, please prepend the column name with the table name, followed by a period. For example: 'borrowers.surname'*

Required

Make sure to update the report group to your library. You may also create a new report subgroup at this time.

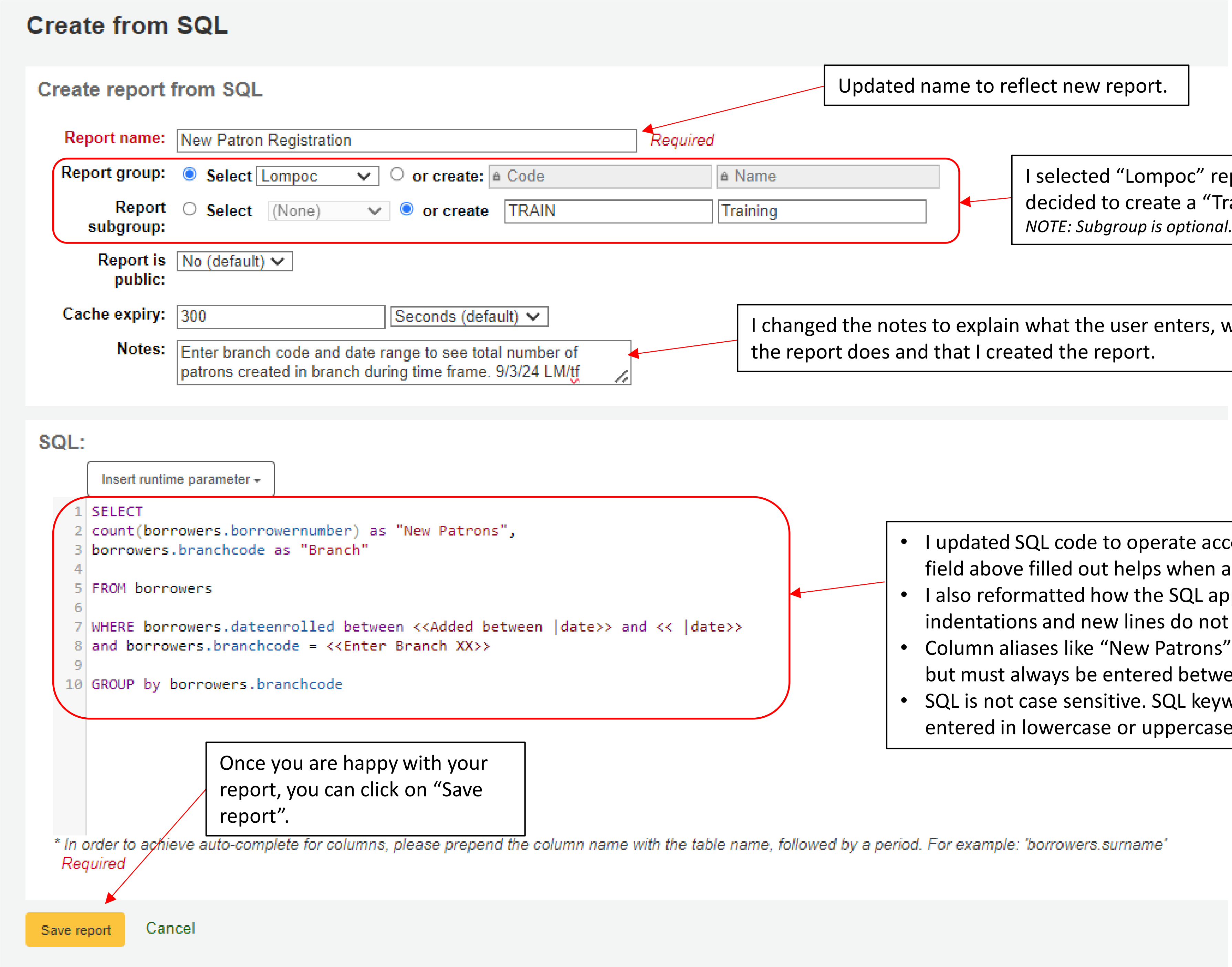
NOTE: If your library does not have the subgroup that is already in the report, you will need to remove the subgroup or create one.

These values do not need to be changed.

Update your notes to reflect what the report will do, what the user must input, name of the author of the report, etc.

Code to be modified for your purpose.

- Once you click on “Duplicate” the new screen will be “Create from SQL”.
- All of the information for the report will be exactly the same as the report that was duplicated.
- You can now edit all of the fields to make the report your own and select “Save report” once you are done.



Create from SQL

Create report from SQL

Report name:	New Patron Registration *Required*
Report group:	● Select Lompoc ○ or create: Code Name
Report subgroup:	○ Select (None) ● or create TRAIN Training
Report is public:	No (default)
Cache expiry:	300 Seconds (default)
Notes:	Enter branch code and date range to see total number of patrons created in branch during time frame. 9/3/24 LM/tf

SQL:

```
SELECT
count(borrowers.borrowernumber) as "New Patrons",
borrowers.branchcode as "Branch"

FROM borrowers

WHERE borrowers.dateenrolled between <<Added between |date>> and << |date>>
and borrowers.branchcode = <<Enter Branch XX>>

GROUP by borrowers.branchcode
```

* *In order to achieve auto-complete for columns, please prepend the column name with the table name, followed by a period. For example: 'borrowers.surname'*

Required

Updated name to reflect new report.

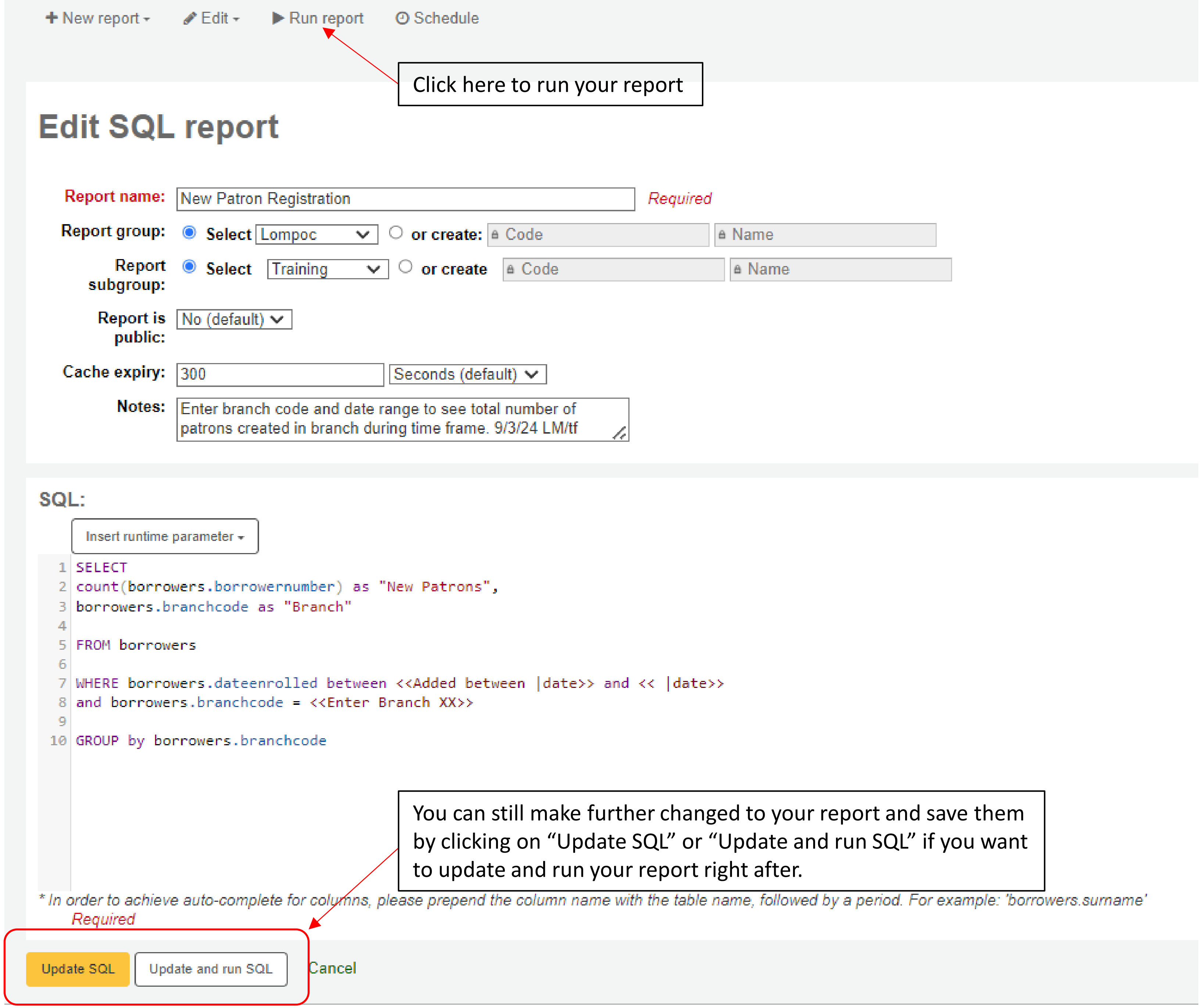
I selected “Lompoc” report group and decided to create a “Training” subgroup.
NOTE: Subgroup is optional.

I changed the notes to explain what the user enters, what the report does and that I created the report.

- I updated SQL code to operate according to my purpose for it. Having the “Notes” field above filled out helps when aligning your code to your purpose.
- I also reformatted how the SQL appeared so it would be easier to read. Spaces, indentations and new lines do not affect how the code is read by the computer.
- Column aliases like “New Patrons” and “Branch” can be changed to your liking, but must always be entered between opening and closing quotation marks.
- SQL is not case sensitive. SQL keywords like “SELECT”, “WHERE” and “and” can be entered in lowercase or uppercase letters.

Once you are happy with your report, you can click on “Save report”.

Your report has been created!
It will now appear in your library’s report tab.



Click here to run your report

Edit SQL report

Report name: New Patron Registration *Required*

Report group: ● Select Lompoc ○ or create: Code Name

Report subgroup: ● Select Training ○ or create Code Name

Report is public: No (default)

Cache expiry: 300 Seconds (default)

Notes: Enter branch code and date range to see total number of patrons created in branch during time frame. 9/3/24 LM/tf

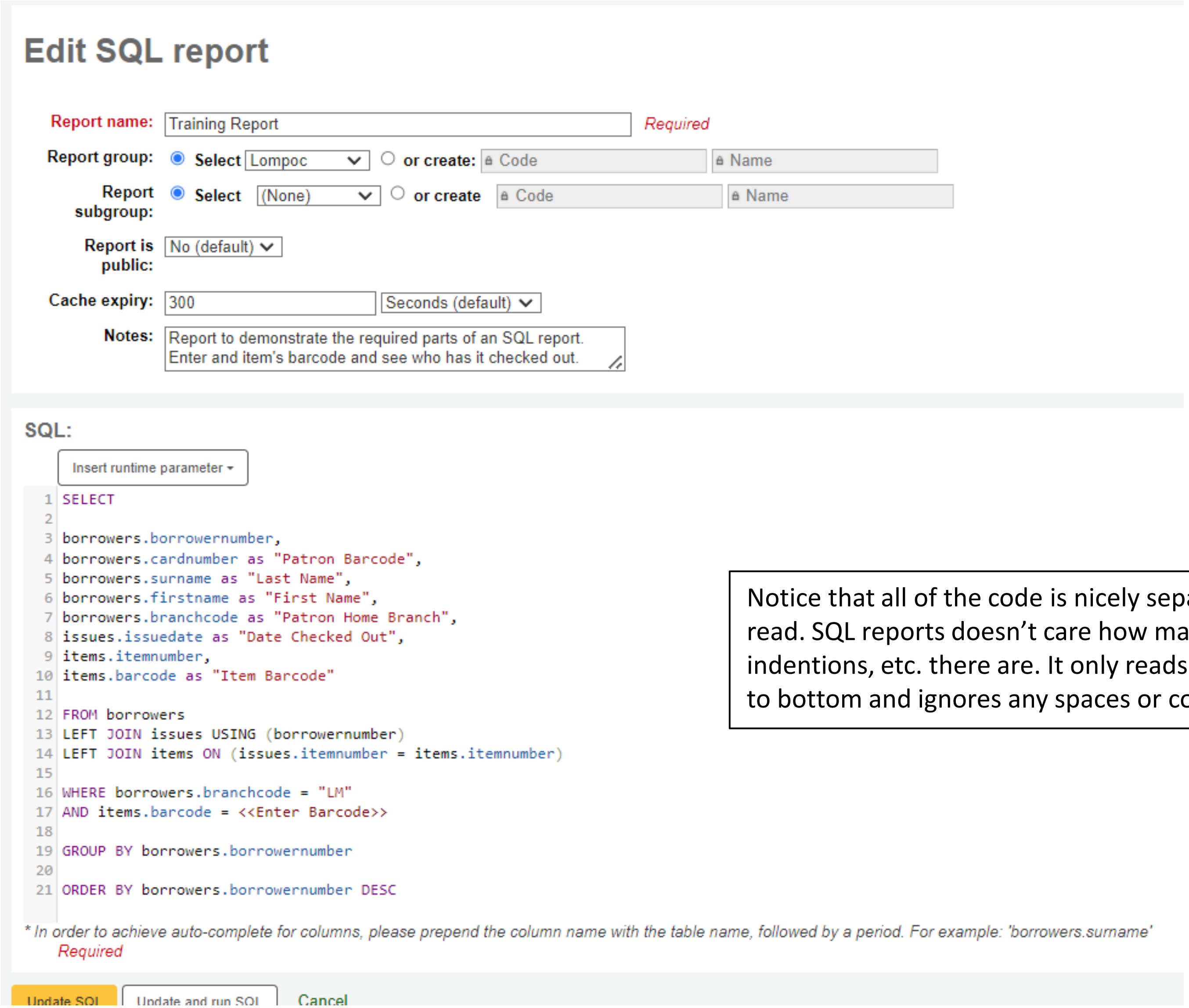
SQL:

```
SELECT
count(borrowers.borrowernumber) as "New Patrons",
borrowers.branchcode as "Branch"

FROM borrowers

WHERE borrowers.dateenrolled between <<Added between |date>> and << |date>>
and borrowers.branchcode = <<Enter Branch XX>>

GROUP by borrowers.branchcode
```

** In order to achieve auto-complete for columns, please prepend the column name with the table name, followed by a period. For example: 'borrowers.surname'*

Required

You can still make further changed to your report and save them by clicking on “Update SQL” or “Update and run SQL” if you want to update and run your report right after.



Edit SQL report

Report name: Training Report *Required*

Report group: ● Select Lompoc ○ or create: Code Name

Report subgroup: ● Select (None) ○ or create Code Name

Report is public: No (default)

Cache expiry: 300 Seconds (default)

Notes: Report to demonstrate the required parts of an SQL report. Enter and item's barcode and see who has it checked out.

SQL:

```
SELECT

borrowers.borrowernumber,
borrowers.cardnumber as "Patron Barcode",
borrowers.surname as "Last Name",
borrowers.firstname as "First Name",
borrowers.branchcode as "Patron Home Branch",
issues.issuedate as "Date Checked Out",
items.itemnumber,
items.barcode as "Item Barcode"

FROM borrowers
LEFT JOIN issues USING (borrowernumber)
LEFT JOIN items ON (issues.itemnumber = items.itemnumber)

WHERE borrowers.branchcode = "LM"
AND items.barcode = <<Enter Barcode>>

GROUP BY borrowers.borrowernumber

ORDER BY borrowers.borrowernumber DESC
```

* *In order to achieve auto-complete for columns, please prepend the column name with the table name, followed by a period. For example: 'borrowers.surname'*
Required

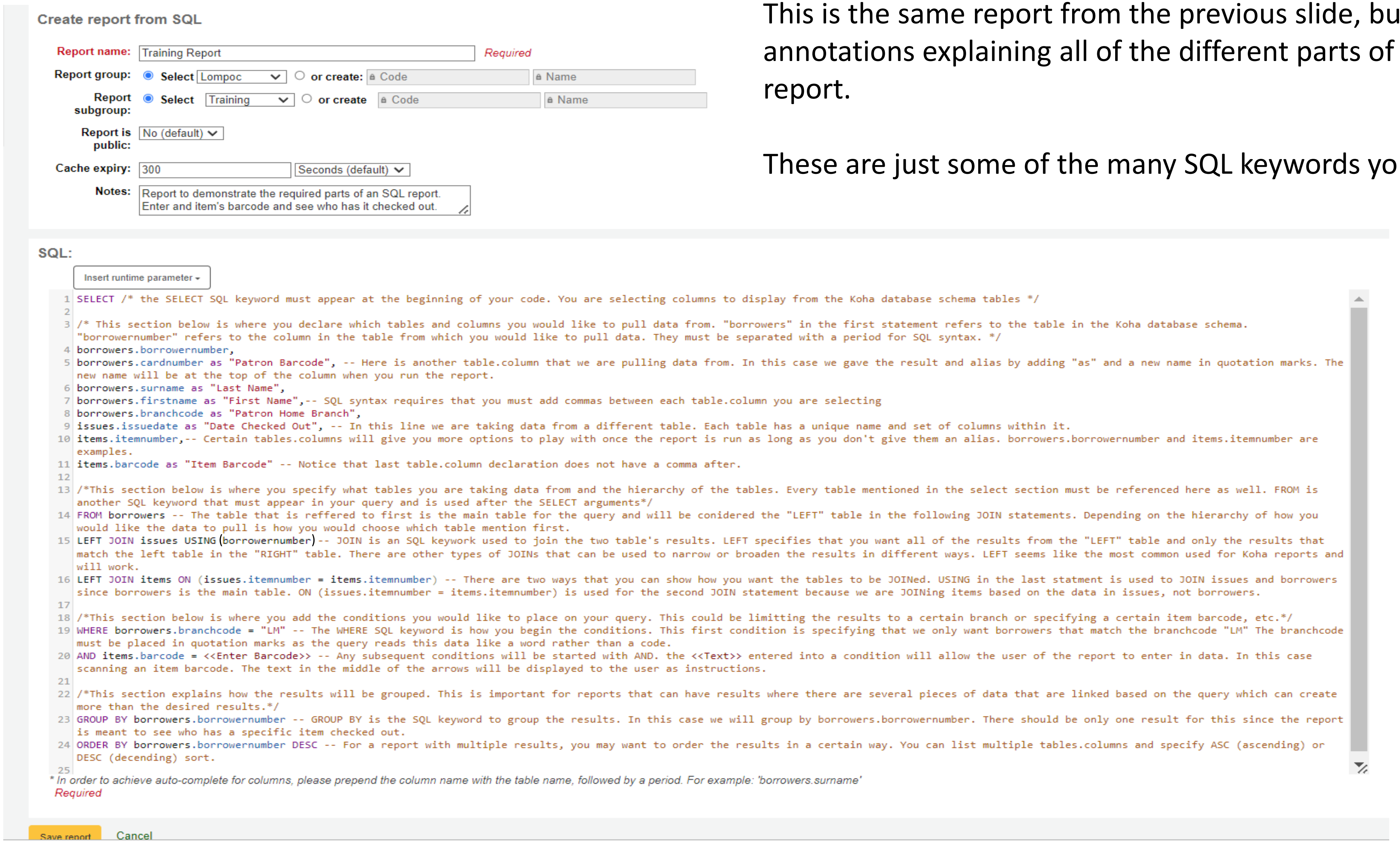
This is a completed report that has all the necessary SQL keywords.

- SELECT
- FROM
- WHERE
- GROUP BY (not absolutely required, but most of the time helpful)

Notice that all of the code is nicely separated and easy to read. SQL reports doesn't care how many spaces, lines indentions, etc. there are. It only reads the code from top to bottom and ignores any spaces or comments.

This is the same report from the previous slide, but with annotations explaining all of the different parts of the report.

These are just some of the many SQL keywords you can use.



Create report from SQL

Report name: Training Report *Required*

Report group: ● Select Lompoc ○ or create: Code Name

Report subgroup: ● Select Training ○ or create Code Name

Report is public: No (default)

Cache expiry: 300 Seconds (default)

Notes: Report to demonstrate the required parts of an SQL report. Enter and item's barcode and see who has it checked out.

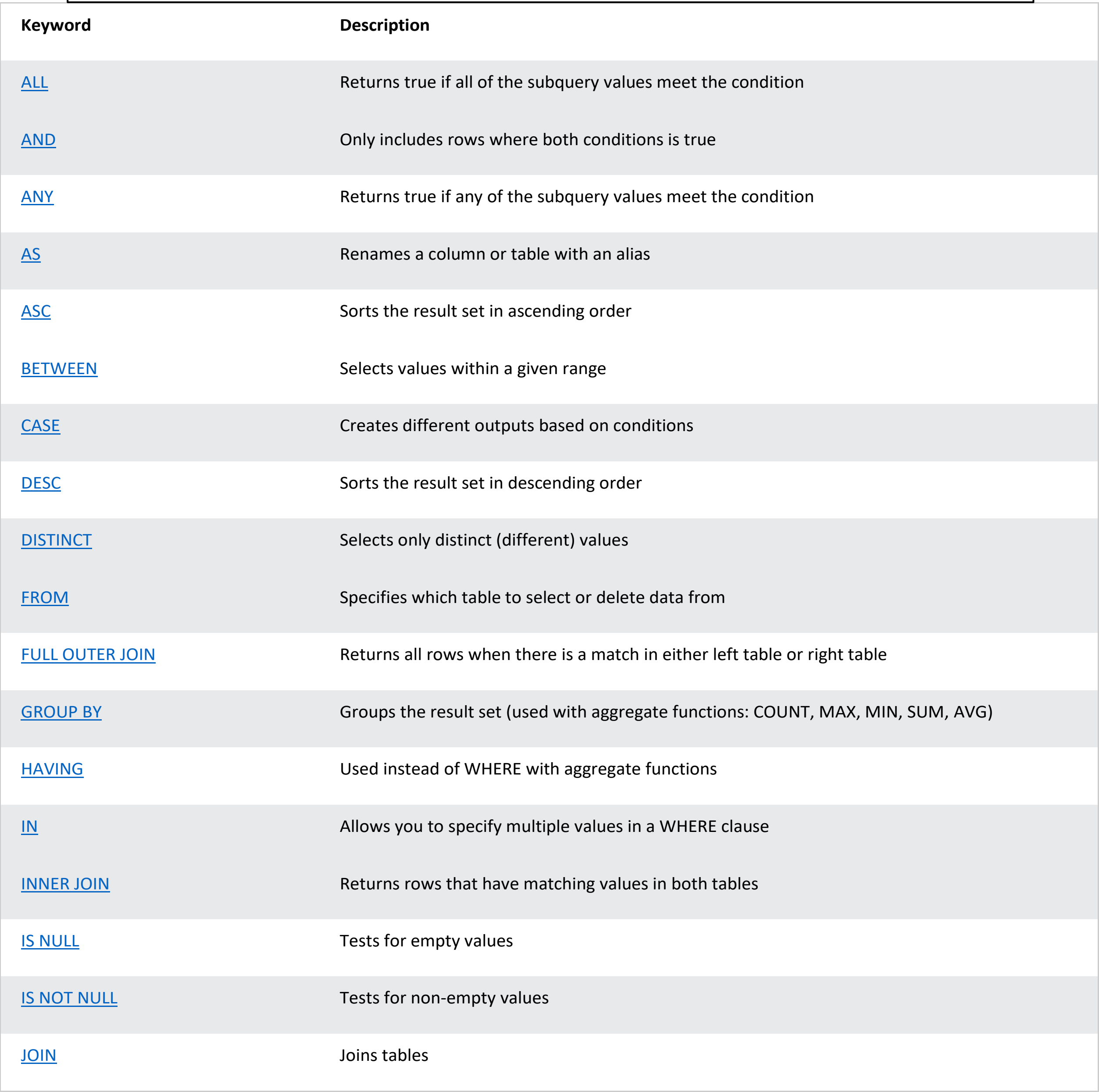
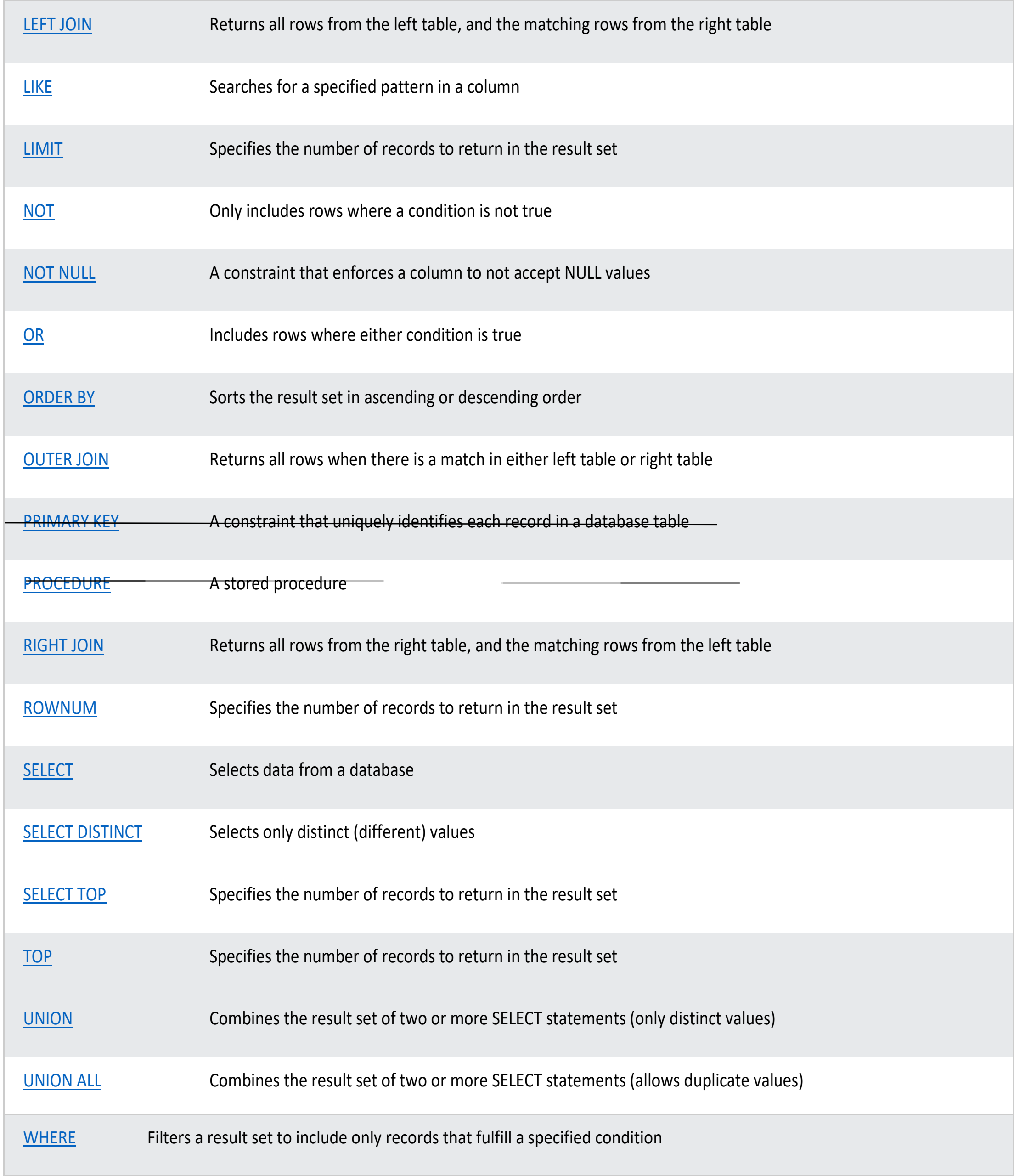
SQL:

```sql
SELECT /* the SELECT SQL keyword must appear at the beginning of your code. You are selecting columns to display from the Koha database schema tables */

/* This section below is where you declare which tables and columns you would like to pull data from. "borrowers" in the first statement refers to the table in the Koha database schema. "borrowernumber" refers to the column in the table from which you would like to pull data. They must be separated with a period for SQL syntax. */
borrowers.borrowernumber,
borrowers.cardnumber as "Patron Barcode", -- Here is another table.column that we are pulling data from. In this case we gave the result and alias by adding "as" and a new name in quotation marks. The new name will be at the top of the column when you run the report.
borrowers.surname as "Last Name",
borrowers.firstname as "First Name",-- SQL syntax requires that you must add commas between each table.column you are selecting
borrowers.branchcode as "Patron Home Branch",
issues.issuedate as "Date Checked Out", -- In this line we are taking data from a different table. Each table has a unique name and set of columns within it.
items.itemnumber,-- Certain tables.columns will give you more options to play with once the report is run as long as you don't give them an alias. borrowers.borrowernumber and items.itemnumber are examples.
items.barcode as "Item Barcode" -- Notice that last table.column declaration does not have a comma after.

/*This section below is where you specify what tables you are taking data from and the hierarchy of the tables. Every table mentioned in the select section must be referenced here as well. FROM is another SQL keyword that must appear in your query and is used after the SELECT arguments*/
FROM borrowers -- The table that is reffered to first is the main table for the query and will be conidered the "LEFT" table in the following JOIN statements. Depending on the hierarchy of how you would like the data to pull is how you would choose which table mention first.
LEFT JOIN issues USING (borrowernumber) -- JOIN is an SQL keywork used to join the two table's results. LEFT specifies that you want all of the results from the "LEFT" table and only the results that match the left table in the "RIGHT" table. There are other types of JOINs that can be used to narrow or broaden the results in different ways. LEFT seems like the most common used for Koha reports and will work.
LEFT JOIN items ON (issues.itemnumber = items.itemnumber) -- There are two ways that you can show how you want the tables to be JOINed. USING in the last statment is used to JOIN issues and borrowers since borrowers is the main table. ON (issues.itemnumber = items.itemnumber) is used for the second JOIN statement because we are JOINing items based on the data in issues, not borrowers.

/*This section below is where you add the conditions you would like to place on your query. This could be limitting the results to a certain branch or specifying a certain item barcode, etc.*/
WHERE borrowers.branchcode = "LM" -- The WHERE SQL keyword is how you begin the conditions. This first condition is specifying that we only want borrowers that match the branchcode "LM" The branchcode must be placed in quotation marks as the query reads this data like a word rather than a code.
AND items.barcode = <<Enter Barcode>> -- Any subsequent conditions will be started with AND. the <<Text>> entered into a condition will allow the user of the report to enter in data. In this case scanning an item barcode. The text in the middle of the arrows will be displayed to the user as instructions.

/*This section explains how the results will be grouped. This is important for reports that can have results where there are several pieces of data that are linked based on the query which can create more than the desired results.*/
GROUP BY borrowers.borrowernumber -- GROUP BY is the SQL keyword to group the results. In this case we will group by borrowers.borrowernumber. There should be only one result for this since the report is meant to see who has a specific item checked out.
ORDER BY borrowers.borrowernumber DESC -- For a report with multiple results, you may want to order the results in a certain way. You can list multiple tables.columns and specify ASC (ascending) or DESC (decending) sort.

```

** In order to achieve auto-complete for columns, please prepend the column name with the table name, followed by a period. For example: 'borrowers.surname'*
Required

SQL Keywords that are accessible in Koha

NOTE: There are more SQL keywords that exist, but Koha has many locked that would change tables, columns and data. Koha reports is “Read Only” in that way.

Keyword	Description
ALL	Returns true if all of the subquery values meet the condition
AND	Only includes rows where both conditions is true
ANY	Returns true if any of the subquery values meet the condition
AS	Renames a column or table with an alias
ASC	Sorts the result set in ascending order
BETWEEN	Selects values within a given range
CASE	Creates different outputs based on conditions
DESC	Sorts the result set in descending order
DISTINCT	Selects only distinct (different) values
FROM	Specifies which table to select or delete data from
FULL OUTER JOIN	Returns all rows when there is a match in either left table or right table
GROUP BY	Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)
HAVING	Used instead of WHERE with aggregate functions
IN	Allows you to specify multiple values in a WHERE clause
INNER JOIN	Returns rows that have matching values in both tables
IS NULL	Tests for empty values
IS NOT NULL	Tests for non-empty values
JOIN	Joins tables
LEFT JOIN	Returns all rows from the left table, and the matching rows from the right table
LIKE	Searches for a specified pattern in a column
LIMIT	Specifies the number of records to return in the result set
NOT	Only includes rows where a condition is not true
NOT NULL	A constraint that enforces a column to not accept NULL values
OR	Includes rows where either condition is true
ORDER BY	Sorts the result set in ascending or descending order
OUTER JOIN	Returns all rows when there is a match in either left table or right table
~~PRIMARY KEY~~	~~A constraint that uniquely identifies each record in a database table~~
~~PROCEDURE~~	~~A stored procedure~~
RIGHT JOIN	Returns all rows from the right table, and the matching rows from the left table
ROWNUM	Specifies the number of records to return in the result set
SELECT	Selects data from a database
SELECT DISTINCT	Selects only distinct (different) values
SELECT TOP	Specifies the number of records to return in the result set
TOP	Specifies the number of records to return in the result set
UNION	Combines the result set of two or more SELECT statements (only distinct values)
UNION ALL	Combines the result set of two or more SELECT statements (allows duplicate values)
WHERE	Filters a result set to include only records that fulfill a specified condition

Resources

- Koha Database Schema of tables and columns https://schema.koha-community.org/23_05/index.html
- Koha Reports Library https://wiki.koha-community.org/wiki/SQL_Reports_Library
- W3 Schools SQL tutorials https://www.w3schools.com/sql/default.asp
- Koha Manual https://koha-community.org/manual/24.05/en/html/reports.html
- ByWater Basic SQL for Koha Users webinar https://bywatersolutions.com/education/basic-sql-for-koha-users
- ByWater Koha Reports articles https://koha.bywatersolutions.com/project/reports
- Code Academy free SQL course https://www.codecademy.com/catalog/language/sql