Video recordings on SQL Labs: https://drive.google.com/drive/folders/1rFz_V3YxlsxE5qrGnIsxpxaTh8bAw8lB?usp=sharing

SQL Commands

Relational databases

SQL queries

SQL filters

SQL joins

Databases

Relational databases in tables. Multiple table

Key

Primary key – unique. One in a table

Foreign key-a column in a table that is a primary key in another table. Dublicate, allows connections to another table together.

Query

A query is a request for data from a database table or a combination of tables.

Applicable in logs analysis.

SQL vs Linux

Purpose

Linux filters data in the context of files and directories on a computer system. It's used for tasks like searching for specific files, manipulating file permissions, or managing processes.

SQL is used to filter data within a database management system. It's used for querying and manipulating data stored in tables and retrieving specific information based on defined criteria.

Syntax

Linux uses various commands and command-line options specific to each filtering tool. Syntax varies depending on the tool and purpose. Some examples of Linux commands are find, sed, cut, e grep

SQL uses the Structured Query Language (SQL), a standardized language with specific keywords and clauses for filtering data across different SQL databases. Some examples of SQL keywords and clauses are WHERE, SELECT, JOIN

Structure

SQL offers a lot more structure than Linux, which is more free-form and not as tidy.

For example, if you wanted to access a log of employee log-in attempts, SQL would have each record separated into columns. Linux would print the data as a line of text without this organization. As a result, selecting a specific column to analyze would be easier and more efficient in SQL.

In terms of structure, SQL provides results that are more easily readable and that can be adjusted more quickly than when using Linux.

Joining tables

Some security-related decisions require information from different tables. SQL allows the analyst to join multiple tables together when returning data. Linux doesn't have that same functionality; it doesn't allow data to be connected to other information on your computer. This is more restrictive for an analyst going through security logs.

Best uses

As a security analyst, it's important to understand when you can use which tool. Although SQL has a more organized structure and allows you to join tables, this doesn't mean that there aren't situations that would require you to filter data in Linux.

A lot of data used in cybersecurity will be stored in a database format that works with SQL. However, other logs might be in a format that is not compatible with SQL. For instance, if the data is stored in a text file, you cannot search through it with SQL. In those cases, it is useful to know how to filter in Linux.

Query

Select and From: not case sensitive

, - for separation

; - at the end of the statement

EG:

```
select employee_ID, device_ID

from employees;

SQL – not case sensitive

*   - select all eg: SELECT *

      From employee;  where * = select all
```

Querry a Database

Select From

```
SELECT customerid, city, country
FROM customers;
```

Order by

```
SELECT customerid, city, country
FROM customers
ORDER BY city;
```

Sorting based on multiple columns

You can also choose multiple columns to order by. For example, you might first choose the `country` and then the `city` column. SQL then sorts the output by `country`, and for rows with the same `country`, it sorts them based on `city`.

```
SELECT customerid, city, country
FROM customers
ORDER BY country, city;
```

Sorting in descending order

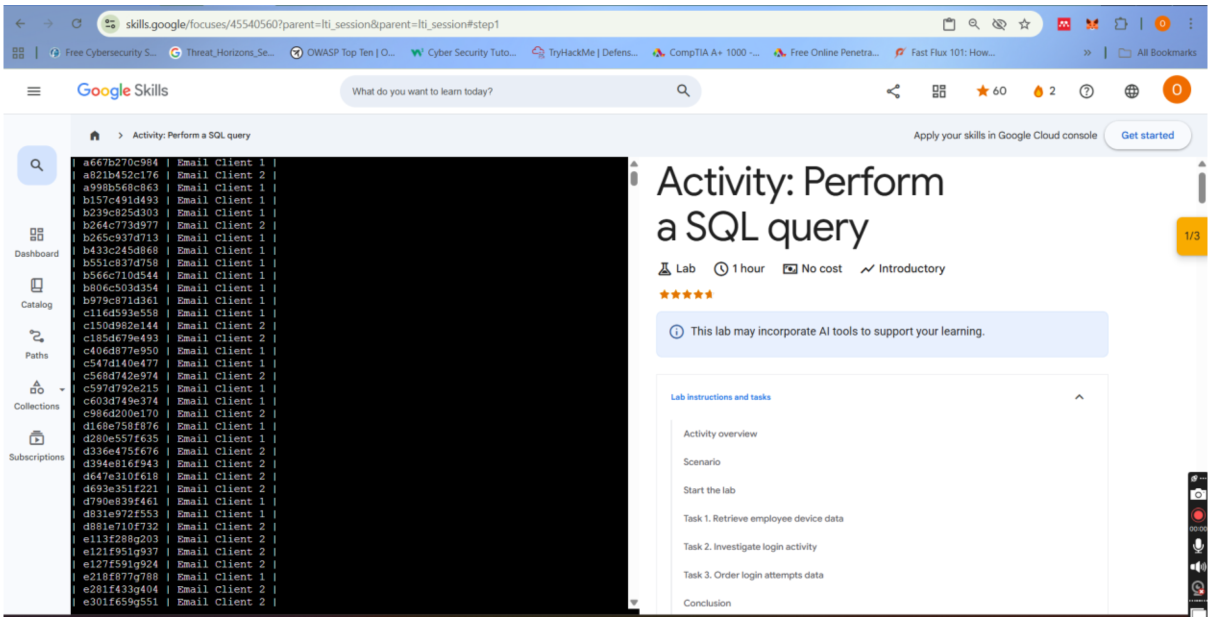
```
SELECT customerid, city, country

FROM customers
ORDER BY city DESC;
```

Lab Practice:

Codes

```
SELECT *
FROM machines;

SELECT device_id, email_client
FROM machines;

SELECT device_id, operating_system, OS_patch_date
FROM machines;

SELECT event_id, country
FROM log_in_attempts;

SELECT username, login_date, login_time
FROM log_in_attempts;

SELECT *
FROM log_in_attempts;

SELECT *
FROM log_in_attempts
ORDER BY login_date;

SELECT *
FROM log_in_attempts
ORDER BY login_date, login_time;
```

…………


skills.google/focuses/45540560?parent=lti_session&parent=lti_session#step1
Google Skills
What do you want to learn today?
Activity: Perform a SQL query
Apply your skills in Google Cloud console
Get started
Dashboard
Catalog
Paths
Collections
Subscriptions
1/3

```
| a667b270c984 | Email Client 1 |
| a821b452c176 | Email Client 2 |
| a998b568c863 | Email Client 1 |
| b157c491d493 | Email Client 1 |
| b239c825d303 | Email Client 1 |
| b264c773d977 | Email Client 2 |
| b265c937d713 | Email Client 1 |
| b433c245d868 | Email Client 1 |
| b551c837d758 | Email Client 1 |
| b566c710d544 | Email Client 1 |
| b806c503d354 | Email Client 1 |
| b979c871d361 | Email Client 1 |
| c116d593e558 | Email Client 1 |
| c150d982e144 | Email Client 2 |
| c185d679e493 | Email Client 2 |
| c406d877e950 | Email Client 1 |
| c547d140e477 | Email Client 1 |
| c568d742e974 | Email Client 2 |
| c597d792e215 | Email Client 1 |
| c603d749e374 | Email Client 1 |
| c986d200e170 | Email Client 2 |
| d168e758f876 | Email Client 1 |
| d280e557f635 | Email Client 1 |
| d336e475f676 | Email Client 2 |
| d394e816f943 | Email Client 2 |
| d647e310f618 | Email Client 2 |
| d693e351f221 | Email Client 2 |
| d790e839f461 | Email Client 1 |
| d831e972f553 | Email Client 1 |
| d881e710f732 | Email Client 2 |
| e113f288g203 | Email Client 2 |
| e121f951g937 | Email Client 2 |
| e127f591g924 | Email Client 2 |
| e218f877g788 | Email Client 1 |
| e281f433g404 | Email Client 2 |
| e301f659g551 | Email Client 2 |
```

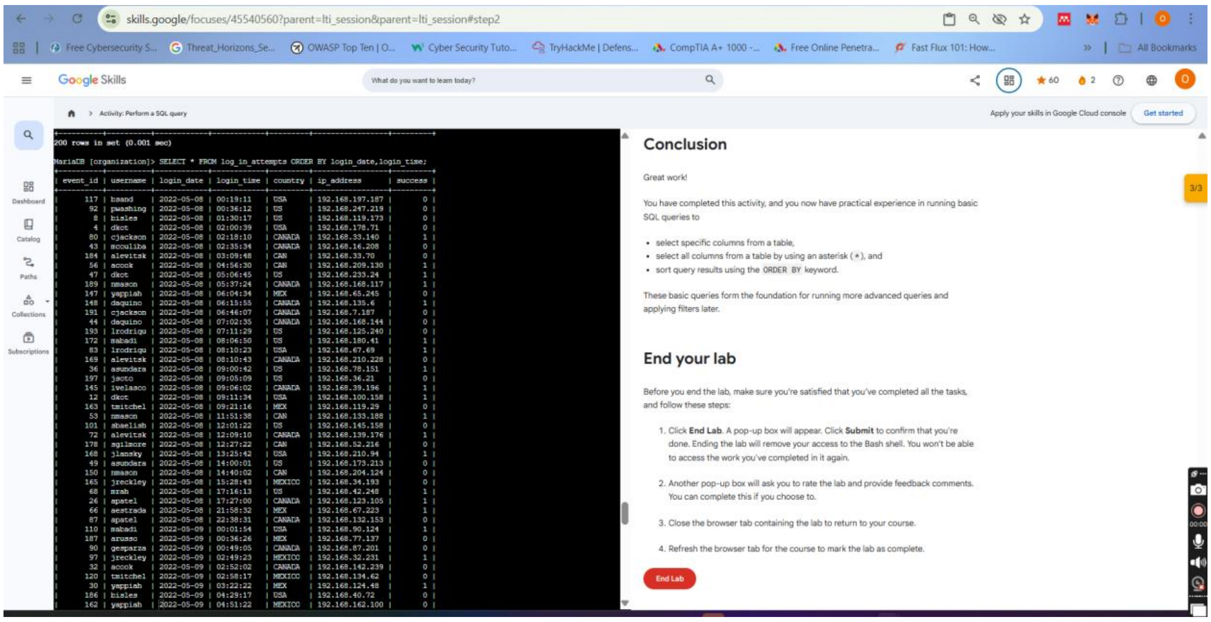
Activity: Perform a SQL query

Lab 1 hour No cost Introductory

This lab may incorporate AI tools to support your learning.

Lab instructions and tasks

- Activity overview
- Scenario
- Start the lab
- Task 1. Retrieve employee device data
- Task 2. Investigate login activity
- Task 3. Order login attempts data
- Conclusion


skills.google/focuses/45540560?parent=lti_session&parent=lti_session#step2
Google Skills
Activity: Perform a SQL query
Get started
3/3
End Lab

```
200 rows in set (0.001 sec)

MariaDB [organization]> SELECT * FROM log_in_attempts ORDER BY login_date,login_time;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|      117 | bsand    | 2022-05-08 | 00:19:11   | USA     | 192.168.197.187 |       0 |
|       92 | pwashing | 2022-05-08 | 00:36:12   | US      | 192.168.247.219 |       0 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
|       80 | cjackson | 2022-05-08 | 02:18:10   | CANADA  | 192.168.33.140  |       1 |
|       43 | mcouliba | 2022-05-08 | 02:35:34   | CANADA  | 192.168.16.208  |       0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70   |       0 |
|       56 | acook    | 2022-05-08 | 04:56:30   | CAN     | 192.168.209.130 |       1 |
|       47 | dkot     | 2022-05-08 | 05:06:45   | US      | 192.168.233.24  |       1 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |       1 |
|      147 | yappiah  | 2022-05-08 | 06:04:34   | MEX     | 192.168.65.245  |       0 |
|      148 | daquino  | 2022-05-08 | 06:15:55   | CANADA  | 192.168.135.6   |       1 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |       0 |
|       44 | daquino  | 2022-05-08 | 07:02:35   | CANADA  | 192.168.168.144 |       0 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |       0 |
|      172 | mabadi   | 2022-05-08 | 08:06:50   | US      | 192.168.180.41  |       1 |
|       83 | lrodriqu | 2022-05-08 | 08:10:23   | USA     | 192.168.67.69   |       1 |
|      169 | alevitsk | 2022-05-08 | 08:10:43   | CANADA  | 192.168.210.228 |       0 |
|       36 | asundara | 2022-05-08 | 09:00:42   | US      | 192.168.78.151  |       1 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |       0 |
|      145 | ivelasco | 2022-05-08 | 09:06:02   | CANADA  | 192.168.39.196  |       1 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
|      163 | tmitchel | 2022-05-08 | 09:21:16   | MEX     | 192.168.119.29  |       0 |
|       53 | nmason   | 2022-05-08 | 11:51:38   | CAN     | 192.168.133.188 |       1 |
|      101 | sbaelish | 2022-05-08 | 12:01:22   | US      | 192.168.145.158 |       0 |
|       72 | alevitsk | 2022-05-08 | 12:09:10   | CANADA  | 192.168.139.176 |       1 |
|      178 | sgilmore | 2022-05-08 | 12:27:22   | CAN     | 192.168.52.216  |       0 |
|      168 | jlansky  | 2022-05-08 | 13:25:42   | USA     | 192.168.210.94  |       1 |
|       49 | asundara | 2022-05-08 | 14:00:01   | US      | 192.168.173.213 |       0 |
|      150 | nmason   | 2022-05-08 | 14:40:02   | CAN     | 192.168.204.124 |       0 |
|      165 | jreckley | 2022-05-08 | 15:28:43   | MEXICO  | 192.168.34.193  |       0 |
|       68 | mrah     | 2022-05-08 | 17:16:13   | US      | 192.168.42.248  |       1 |
|       26 | apatel   | 2022-05-08 | 17:27:00   | CANADA  | 192.168.123.105 |       1 |
|       66 | aestrada | 2022-05-08 | 21:58:32   | MEX     | 192.168.67.223  |       1 |
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
|      110 | mabadi   | 2022-05-09 | 00:01:54   | USA     | 192.168.90.124  |       1 |
|      187 | arusso   | 2022-05-09 | 00:36:26   | MEX     | 192.168.77.137  |       0 |
|       90 | gesparza | 2022-05-09 | 00:49:05   | CANADA  | 192.168.87.201  |       0 |
|       97 | jreckley | 2022-05-09 | 02:49:23   | MEXICO  | 192.168.32.231  |       1 |
|       32 | acook    | 2022-05-09 | 02:52:02   | CANADA  | 192.168.142.239 |       0 |
|      120 | tmitchel | 2022-05-09 | 02:58:17   | MEXICO  | 192.168.134.62  |       0 |
|       30 | yappiah  | 2022-05-09 | 03:22:22   | MEX     | 192.168.124.48  |       1 |
|      186 | bisles   | 2022-05-09 | 04:29:17   | USA     | 192.168.40.72   |       0 |
|      162 | yappiah  | 2022-05-09 | 04:51:22   | MEXICO  | 192.168.162.100 |       0 |
```

Conclusion

Great work!

You have completed this activity, and you now have practical experience in running basic SQL queries to

- select specific columns from a table,
- select all columns from a table by using an asterisk (`*`), and
- sort query results using the `ORDER BY` keyword.

These basic queries form the foundation for running more advanced queries and applying filters later.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.

Filtering

Where = eqial sign is used for filtering.

```
SELECT *
	FROM log_in_attempts
	WHERE country = ‘USA’;
```

#For pattern. %, used together with the word LIKE instead of = sign

```
SELECT *
	FROM log_in_attempts
	WHERE country LIKE ‘US%’;
```

Filtering for patterns

You can also filter based on a pattern. For example, you can identify entries that start or end with a certain character or characters. Filtering for a pattern requires incorporating two more elements into your `WHERE` clause:

- a wildcard
- the `LIKE` operator

Wildcards

A **wildcard** is a special character that can be substituted with any other character. Two of the most useful wildcards are the percentage sign (`%`) and the underscore (`_`):

- The percentage sign substitutes for any number of other characters.
- The underscore symbol only substitutes for one other character.

These wildcards can be placed after a string, before a string, or in both locations depending on the pattern you're filtering for.

The following table includes these wildcards applied to the string `'a'` and examples of what each pattern would return.

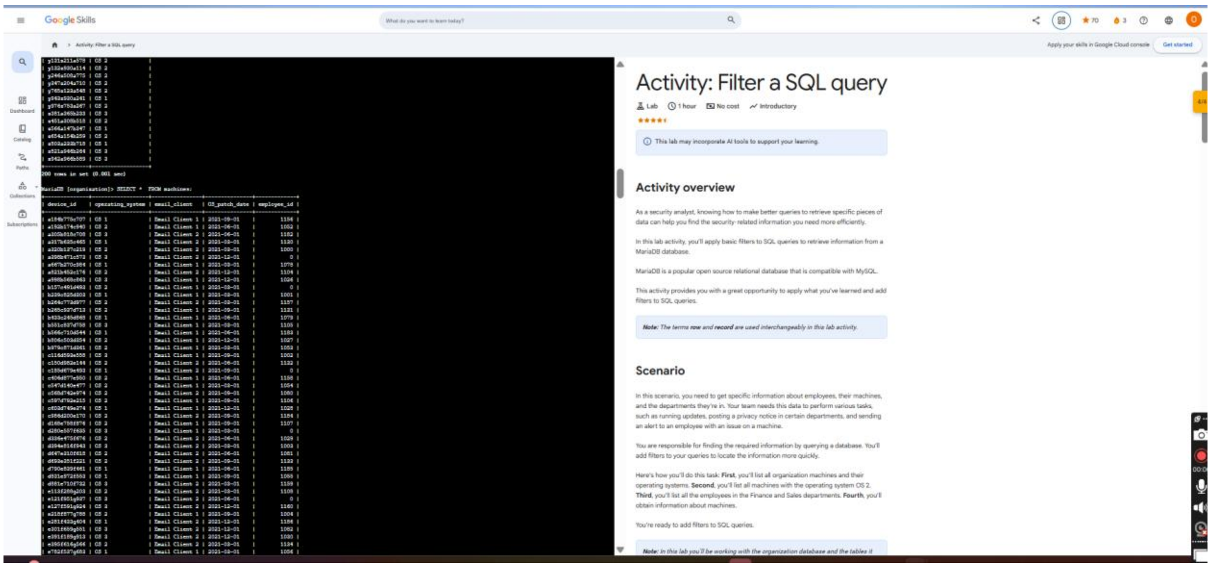
Pattern	Results that could be returned
`'a%'`	`apple123, art, a`
`'a_'`	`as, an, a7`
`'a__'`	`ant, add, a1c`
`'%a'`	`pizza, Z6ra, a`
`'_a'`	`ma, 1a, Ha`

Pattern	Results that could be returned
`'%a%'`	`Again, back, a`
`'_a_'`	`Car, ban, ea7`

Practical

Code

```
SELECT device_id, operating_system
FROM machines;

SELECT device_id, operating_system
FROM machines
WHERE operating_system = 'OS 2';

SELECT *
FROM employees
WHERE department = 'Finance';

SELECT *
FROM employees
WHERE department = 'Sales';

SELECT *
FROM employees
WHERE office = 'South-109';

SELECT *
FROM employees
WHERE office LIKE 'South%';
```



Activity: Filter a SQL query

```
| y121a211a578 | OS 2             |
| y132a930a114 | OS 2             |
| y246a508a775 | OS 2             |
| y247a204a710 | OS 2             |
| y765a122a548 | OS 2             |
| y943z930a241 | OS 1             |
| y976z753a267 | OS 2             |
| z281a365b233 | OS 3             |
| z451a308b518 | OS 2             |
| z566a147b247 | OS 1             |
| z654a154b259 | OS 2             |
| z803a233b718 | OS 1             |
| z821a946b264 | OS 3             |
| z942a966b589 | OS 3             |
+--------------+------------------+
200 rows in set (0.001 sec)

MariaDB [organization]> SELECT *  FROM machines;
+--------------+------------------+----------------+---------------+-------------+
| device_id    | operating_system | email_client   | OS_patch_date | employee_id |
+--------------+------------------+----------------+---------------+-------------+
| a184b775c707 | OS 1             | Email Client 1 | 2021-09-01    |        1156 |
| a192b174c940 | OS 2             | Email Client 1 | 2021-06-01    |        1052 |
| a305b818c708 | OS 3             | Email Client 2 | 2021-06-01    |        1182 |
| a317b635c465 | OS 1             | Email Client 2 | 2021-03-01    |        1120 |
| a320b137c219 | OS 2             | Email Client 2 | 2021-03-01    |        1000 |
| a398b471c573 | OS 3             | Email Client 2 | 2021-12-01    |           0 |
| a667b270c984 | OS 1             | Email Client 1 | 2021-03-01    |        1078 |
| a821b452c176 | OS 2             | Email Client 2 | 2021-12-01    |        1104 |
| a998b568c863 | OS 3             | Email Client 1 | 2021-12-01    |        1026 |
| b157c491d492 | OS 2             | Email Client 1 | 2021-03-01    |           0 |
| b239c825d303 | OS 1             | Email Client 1 | 2021-03-01    |        1001 |
| b264c773d977 | OS 2             | Email Client 2 | 2021-03-01    |        1157 |
| b265c937d713 | OS 2             | Email Client 1 | 2021-09-01    |        1131 |
| b433c245d565 | OS 1             | Email Client 1 | 2021-06-01    |        1079 |
| b551c837d758 | OS 3             | Email Client 1 | 2021-03-01    |        1105 |
| b566c710d544 | OS 1             | Email Client 1 | 2021-06-01    |        1183 |
| b806c503d354 | OS 2             | Email Client 1 | 2021-12-01    |        1027 |
| b879c871d261 | OS 2             | Email Client 1 | 2021-03-01    |        1053 |
| c116d593e558 | OS 3             | Email Client 1 | 2021-09-01    |        1002 |
| c150d982e144 | OS 2             | Email Client 2 | 2021-06-01    |        1132 |
| c185d679e493 | OS 1             | Email Client 2 | 2021-09-01    |           0 |
| c406d877e950 | OS 2             | Email Client 1 | 2021-06-01    |        1158 |
| c547d140e477 | OS 2             | Email Client 1 | 2021-03-01    |        1054 |
| c568d742e974 | OS 2             | Email Client 2 | 2021-09-01    |        1080 |
| c597d792e215 | OS 2             | Email Client 1 | 2021-09-01    |        1106 |
| c602d749e274 | OS 1             | Email Client 1 | 2021-12-01    |        1028 |
| c986d200e170 | OS 2             | Email Client 2 | 2021-09-01    |        1184 |
| d168e758f576 | OS 2             | Email Client 1 | 2021-09-01    |        1107 |
| d280e557f635 | OS 3             | Email Client 1 | 2021-03-01    |           0 |
| d336e475f676 | OS 2             | Email Client 2 | 2021-06-01    |        1029 |
| d394e816f943 | OS 3             | Email Client 2 | 2021-03-01    |        1003 |
| d647e310f618 | OS 2             | Email Client 2 | 2021-06-01    |        1081 |
| d692e251f221 | OS 2             | Email Client 2 | 2021-09-01    |        1133 |
| d790e829f461 | OS 1             | Email Client 1 | 2021-06-01    |        1185 |
| d831e972f563 | OS 1             | Email Client 1 | 2021-09-01    |        1055 |
| d881e710f732 | OS 3             | Email Client 2 | 2021-03-01    |        1159 |
| e113f289g203 | OS 2             | Email Client 2 | 2021-03-01    |        1108 |
| e121f851g927 | OS 3             | Email Client 2 | 2021-06-01    |           0 |
| e127f591g924 | OS 3             | Email Client 2 | 2021-12-01    |        1160 |
| e218f877g788 | OS 2             | Email Client 1 | 2021-09-01    |        1004 |
| e281f423g404 | OS 1             | Email Client 2 | 2021-12-01    |        1186 |
| e301f689g881 | OS 3             | Email Client 2 | 2021-12-01    |        1082 |
| e391f189g913 | OS 3             | Email Client 2 | 2021-12-01    |        1030 |
| e395f616g566 | OS 2             | Email Client 2 | 2021-03-01    |        1134 |
| e782f527g682 | OS 1             | Email Client 1 | 2021-03-01    |        1056 |
```

Activity: Filter a SQL query

Lab · 1 hour · No cost · Introductory

This lab may incorporate AI tools to support your learning.

Activity overview

As a security analyst, knowing how to make better queries to retrieve specific pieces of data can help you find the security-related information you need more efficiently.

In this lab activity, you'll apply basic filters to SQL queries to retrieve information from a MariaDB database.

MariaDB is a popular open source relational database that is compatible with MySQL.

This activity provides you with a great opportunity to apply what you've learned and add filters to SQL queries.

***Note:** The terms **row** and **record** are used interchangeably in this lab activity.*

Scenario

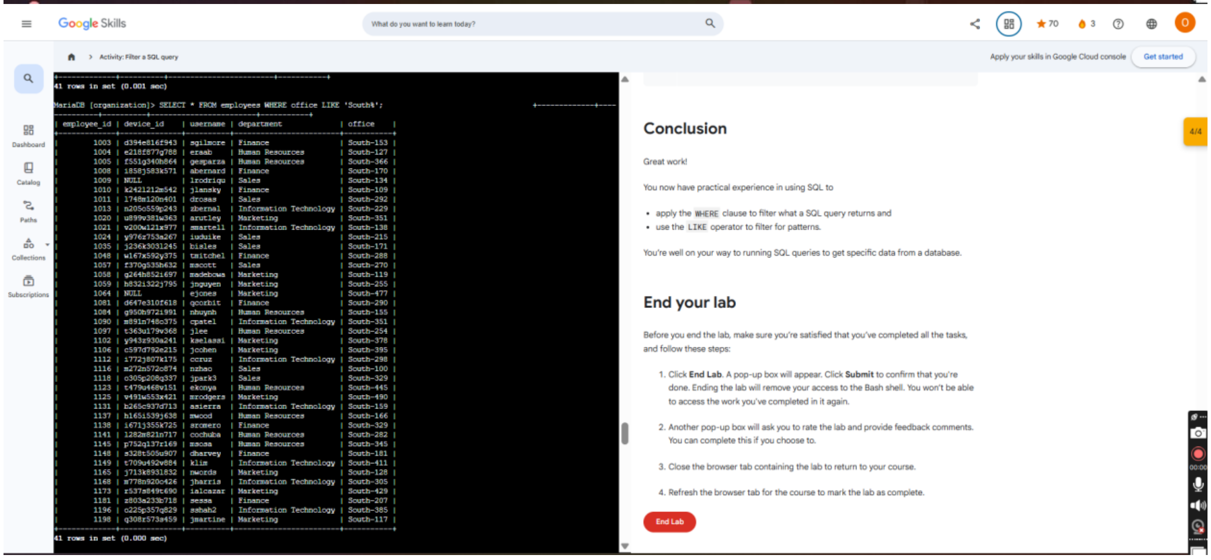
In this scenario, you need to get specific information about employees, their machines, and the departments they're in. Your team needs this data to perform various tasks, such as running updates, posting a privacy notice in certain departments, and sending an alert to an employee with an issue on a machine.

You are responsible for finding the required information by querying a database. You'll add filters to your queries to locate the information more quickly.

Here's how you'll do this task: **First**, you'll list all organization machines and their operating systems. **Second**, you'll list all machines with the operating system OS 2. **Third**, you'll list all the employees in the Finance and Sales departments. **Fourth**, you'll obtain information about machines.

You're ready to add filters to SQL queries.

***Note:** In this lab you'll be working with the organization database and the tables it*



```
+-------------+--------------+-----------+------------------------+-----------+
41 rows in set (0.001 sec)

MariaDB [organization]> SELECT * FROM employees WHERE office LIKE 'South%';
+-------------+--------------+-----------+------------------------+-----------+
| employee_id | device_id    | username  | department             | office    |
+-------------+--------------+-----------+------------------------+-----------+
|        1003 | d394e816f943 | sgilmore  | Finance                | South-153 |
|        1004 | e218f877g788 | eraab     | Human Resources        | South-127 |
|        1005 | f551g340h864 | gesparza  | Human Resources        | South-366 |
|        1008 | i858j583k571 | abernard  | Finance                | South-170 |
|        1009 | NULL         | lrodriqu  | Sales                  | South-134 |
|        1010 | k2421212m542 | jlansky   | Finance                | South-109 |
|        1011 | 1748m120n401 | drosas    | Sales                  | South-292 |
|        1013 | n205o559p243 | zbernal   | Information Technology | South-229 |
|        1020 | u899v381w363 | arutley   | Marketing              | South-351 |
|        1021 | v200w121x977 | smartell  | Information Technology | South-138 |
|        1024 | y976z753a267 | iuduike   | Sales                  | South-215 |
|        1035 | j236k3031245 | bisles    | Sales                  | South-171 |
|        1048 | w167x592y375 | tmitchel  | Finance                | South-288 |
|        1057 | f370g535h632 | mscott    | Sales                  | South-270 |
|        1058 | g264h852i697 | madebowa  | Marketing              | South-119 |
|        1059 | h832i322j795 | jnguyen   | Marketing              | South-255 |
|        1064 | NULL         | ejones    | Marketing              | South-477 |
|        1081 | d647e310f618 | gcorbit   | Finance                | South-290 |
|        1084 | g950h972i991 | nhuynh    | Human Resources        | South-155 |
|        1090 | m891n748o375 | cpatel    | Information Technology | South-351 |
|        1097 | t363u179v368 | jlee      | Human Resources        | South-254 |
|        1102 | y943z930a241 | kselassi  | Marketing              | South-378 |
|        1106 | c597d792e215 | jcohen    | Marketing              | South-395 |
|        1112 | i772j807k175 | ccruz     | Information Technology | South-298 |
|        1116 | m272n572o874 | nzhao     | Sales                  | South-100 |
|        1118 | o305p208q337 | jpark3    | Sales                  | South-329 |
|        1123 | t479u468v151 | ekonya    | Human Resources        | South-445 |
|        1125 | v491w553x421 | mrodgers  | Marketing              | South-490 |
|        1131 | b265c937d713 | asierra   | Information Technology | South-159 |
|        1137 | h165i539j638 | mwood     | Human Resources        | South-166 |
|        1138 | i671j355k725 | sromero   | Finance                | South-329 |
|        1141 | 1282m821n717 | cochuba   | Human Resources        | South-282 |
|        1145 | p752q137r169 | msosa     | Human Resources        | South-345 |
|        1148 | s328t505u907 | dharvey   | Finance                | South-181 |
|        1149 | t709u492v884 | klim      | Information Technology | South-411 |
|        1165 | j713k8931832 | nwords    | Marketing              | South-128 |
|        1168 | m778n920o426 | jharris   | Information Technology | South-305 |
|        1173 | r537s849t690 | ialcazar  | Marketing              | South-429 |
|        1181 | z803a233b718 | sessa     | Finance                | South-207 |
|        1196 | o225p357q829 | sshah2    | Information Technology | South-385 |
|        1198 | q308r573s459 | jmartine  | Marketing              | South-117 |
+-------------+--------------+-----------+------------------------+-----------+
41 rows in set (0.000 sec)
```

Conclusion

Great work!

You now have practical experience in using SQL to

- apply the `WHERE` clause to filter what a SQL query returns and
- use the `LIKE` operator to filter for patterns.

You're well on your way to running SQL queries to get specific data from a database.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.

Filters on numeric and date and time data:

Comparison operators

In SQL, filtering numeric and date and time data often involves operators. You can use the following operators in your filters to make sure you return only the rows you need:

operator	use
<	less than

operator	use
`>`	greater than
`=`	equal to
`<=`	less than or equal to
`>=`	greater than or equal to
`<>`	not equal to

Note: You can also use `!=` as an alternative operator for not equal to.

NOTE: for date and time, use quotation marks but not the case in numbers

In other words, the `>` operator is exclusive and the `>=` operator is inclusive. An **exclusive operator** is an operator that does not include the value of comparison. An **inclusive operator** is an operator that includes the value of comparison.

BETWEEN

`BETWEEN` filters for numbers or dates within a range. For example, if you want to find the first and last names of all employees hired between January 1, 2002 and January 1, 2003, you can use the `BETWEEN` operator as follows:

Reset

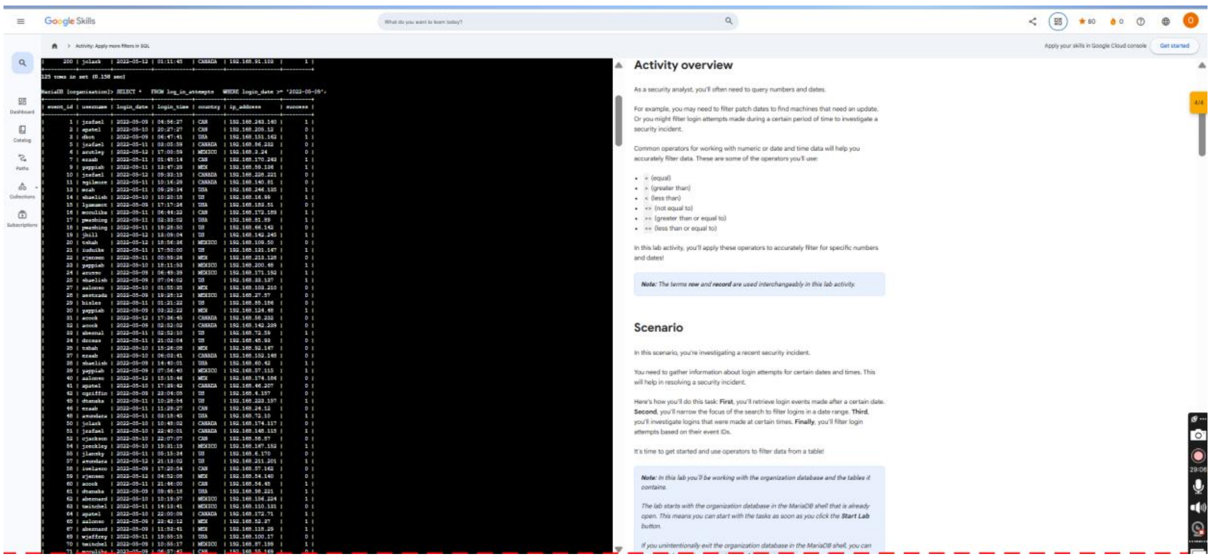
Note: The `BETWEEN` operator is inclusive. This means records with a `hiredate` of January 1, 2002 or January 1, 2003 are included in the results of the previous query.

Lab Practice

Codes used

```
SELECT  *
```

```
FROM log_in_attempts
WHERE login_date > '2023-01-15';

SELECT *
FROM log_in_attempts
WHERE login_date BETWEEN '2023-02-01' AND '2023-02-07';

SELECT *
FROM log_in_attempts
WHERE login_time = '09:30:00';

SELECT *
FROM log_in_attempts
WHERE login_id = 503;
```


Google Skills
What do you want to learn today?
Activity: Apply more filters in SQL
Dashboard
Catalog
Paths
Collections
Subscriptions
Apply your skills in Google Cloud console
Get started

```
|      200 | jclark   | 2022-05-12 | 01:11:45   | CANADA  | 192.168.91.102  |       1 |
+----------+----------+------------+------------+---------+-----------------+---------+
125 rows in set (0.158 sec)

MariaDB [organization]> SELECT *  FROM log_in_attempts  WHERE login_date >= '2022-05-09';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
|        6 | arutley  | 2022-05-12 | 17:00:59   | MEXICO  | 192.168.3.24    |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
|        9 | yappiah  | 2022-05-11 | 13:47:29   | MEX     | 192.168.59.136  |       1 |
|       10 | jrafael  | 2022-05-12 | 09:33:19   | CANADA  | 192.168.228.221 |       0 |
|       11 | sgilmore | 2022-05-11 | 10:16:29   | CANADA  | 192.168.140.81  |       0 |
|       13 | mrah     | 2022-05-11 | 09:29:34   | USA     | 192.168.246.135 |       1 |
|       14 | abellmas | 2022-05-10 | 10:20:18   | US      | 192.168.16.99   |       1 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
|       16 | mcouliba | 2022-05-11 | 06:44:22   | CAN     | 192.168.172.189 |       1 |
|       17 | pwashing | 2022-05-11 | 02:33:02   | USA     | 192.168.81.89   |       1 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142  |       0 |
|       19 | jhill    | 2022-05-12 | 13:09:04   | US      | 192.168.142.245 |       1 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50  |       0 |
|       21 | iuduike  | 2022-05-11 | 17:50:00   | US      | 192.168.131.147 |       1 |
|       22 | rjensen  | 2022-05-11 | 00:59:26   | MEX     | 192.168.213.128 |       0 |
|       23 | yappiah  | 2022-05-10 | 18:11:53   | MEXICO  | 192.168.200.48  |       1 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       25 | abellmas | 2022-05-09 | 07:04:02   | US      | 192.168.33.137  |       1 |
|       27 | aalonso  | 2022-05-10 | 01:55:35   | MEX     | 192.168.103.210 |       0 |
|       28 | aestrada | 2022-05-09 | 19:25:12   | MEXICO  | 192.168.27.57   |       0 |
|       29 | bisles   | 2022-05-11 | 01:21:22   | US      | 192.168.85.186  |       0 |
|       30 | yappiah  | 2022-05-09 | 03:22:22   | MEX     | 192.168.124.48  |       1 |
|       31 | acook    | 2022-05-12 | 17:36:45   | CANADA  | 192.168.58.232  |       0 |
|       32 | acook    | 2022-05-09 | 02:52:02   | CANADA  | 192.168.142.239 |       0 |
|       33 | abernal  | 2022-05-11 | 02:52:10   | US      | 192.168.72.59   |       1 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
|       35 | tshah    | 2022-05-10 | 15:26:08   | MEX     | 192.168.92.147  |       0 |
|       37 | eraab    | 2022-05-10 | 06:03:41   | CANADA  | 192.168.152.148 |       0 |
|       38 | abellmas | 2022-05-09 | 14:40:01   | USA     | 192.168.60.42   |       1 |
|       39 | yappiah  | 2022-05-09 | 07:56:40   | MEXICO  | 192.168.57.115  |       1 |
|       40 | aalonso  | 2022-05-12 | 15:15:46   | MEX     | 192.168.174.186 |       0 |
|       41 | apatel   | 2022-05-10 | 17:39:42   | CANADA  | 192.168.46.207  |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
|       45 | dtanaka  | 2022-05-11 | 10:28:54   | US      | 192.168.223.157 |       1 |
|       46 | eraab    | 2022-05-11 | 11:29:27   | CAN     | 192.168.24.12   |       0 |
|       48 | asundara | 2022-05-11 | 03:18:45   | USA     | 192.168.72.10   |       1 |
|       50 | jclark   | 2022-05-10 | 10:48:02   | CANADA  | 192.168.174.117 |       0 |
|       51 | jrafael  | 2022-05-10 | 22:40:01   | CANADA  | 192.168.148.115 |       1 |
|       52 | cjackson | 2022-05-10 | 22:07:07   | CAN     | 192.168.58.57   |       0 |
|       54 | jreckley | 2022-05-10 | 13:31:19   | MEXICO  | 192.168.167.152 |       1 |
|       55 | jlansky  | 2022-05-11 | 08:15:34   | US      | 192.168.6.170   |       0 |
|       57 | asundara | 2022-05-12 | 21:13:02   | US      | 192.168.211.201 |       1 |
|       58 | ivelasco | 2022-05-09 | 17:20:54   | CAN     | 192.168.57.162  |       0 |
|       59 | rjensen  | 2022-05-12 | 04:52:08   | MEX     | 192.168.54.140  |       0 |
|       60 | acook    | 2022-05-11 | 21:46:00   | CAN     | 192.168.54.45   |       1 |
|       61 | dtanaka  | 2022-05-09 | 09:45:18   | USA     | 192.168.98.221  |       1 |
|       62 | abernard | 2022-05-10 | 10:19:57   | MEXICO  | 192.168.156.224 |       1 |
|       63 | tmitchel | 2022-05-11 | 14:13:41   | MEXICO  | 192.168.110.131 |       0 |
|       64 | apatel   | 2022-05-10 | 22:00:09   | CANADA  | 192.168.172.71  |       1 |
|       65 | aalonso  | 2022-05-09 | 22:42:12   | MEX     | 192.168.52.37   |       1 |
|       67 | abernard | 2022-05-09 | 11:53:41   | MEX     | 192.168.118.29  |       1 |
|       69 | wjaffrey | 2022-05-11 | 19:55:15   | USA     | 192.168.100.17  |       0 |
|       70 | tmitchel | 2022-05-09 | 10:56:17   | MEXICO  | 192.168.87.199  |       1 |
|       71 | mcouliba | 2022-05-09 | 06:57:42   | CAN     | 192.168.55.169  |       0 |
```

Activity overview

As a security analyst, you'll often need to query numbers and dates.

For example, you may need to filter patch dates to find machines that need an update. Or you might filter login attempts made during a certain period of time to investigate a security incident.

Common operators for working with numeric or date and time data will help you accurately filter data. These are some of the operators you'll use:

- `=` (equal)
- `>` (greater than)
- `<` (less than)
- `<>` (not equal to)
- `>=` (greater than or equal to)
- `<=` (less than or equal to)

In this lab activity, you'll apply these operators to accurately filter for specific numbers and dates!

***Note:** The terms **row** and **record** are used interchangeably in this lab activity.*

Scenario

In this scenario, you're investigating a recent security incident.

You need to gather information about login attempts for certain dates and times. This will help in resolving a security incident.

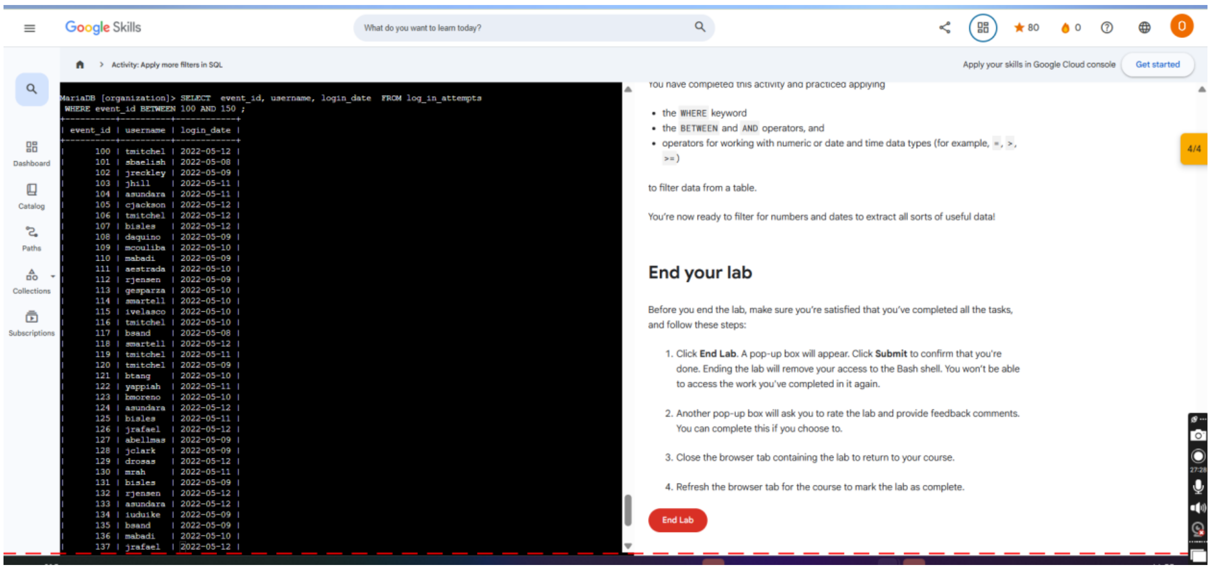
Here's how you'll do this task: **First**, you'll retrieve login events made after a certain date. **Second**, you'll narrow the focus of the search to filter logins in a date range. **Third**, you'll investigate logins that were made at certain times. **Finally**, you'll filter login attempts based on their event IDs.

It's time to get started and use operators to filter data from a table!

***Note:** In this lab you'll be working with the organization database and the tables it contains.*

*The lab starts with the organization database in the MariaDB shell that is already open. This means you can start with the tasks as soon as you click the **Start Lab** button.*

If you unintentionally exit the organization database in the MariaDB shell, you can


Google Skills
What do you want to learn today?
80
0
Activity: Apply more filters in SQL
Apply your skills in Google Cloud console
Get started
Dashboard
Catalog
Paths
Collections
Subscriptions
4/4
End Lab

```
MariaDB [organization]> SELECT  event_id, username, login_date  FROM log_in_attempts
 WHERE event_id BETWEEN 100 AND 150 ;
+----------+----------+------------+
| event_id | username | login_date |
+----------+----------+------------+
|      100 | tmitchel | 2022-05-12 |
|      101 | abellmas | 2022-05-08 |
|      102 | jreckley | 2022-05-09 |
|      103 | jhill    | 2022-05-11 |
|      104 | asundara | 2022-05-11 |
|      105 | cjackson | 2022-05-12 |
|      106 | tmitchel | 2022-05-12 |
|      107 | bisles   | 2022-05-12 |
|      108 | daquino  | 2022-05-09 |
|      109 | mcouliba | 2022-05-10 |
|      110 | mabadi   | 2022-05-09 |
|      111 | aestrada | 2022-05-10 |
|      112 | rjensen  | 2022-05-09 |
|      113 | gesparza | 2022-05-10 |
|      114 | asmartell| 2022-05-09 |
|      115 | ivelasco | 2022-05-10 |
|      116 | tmitchel | 2022-05-10 |
|      117 | bsand    | 2022-05-08 |
|      118 | asmartell| 2022-05-12 |
|      119 | tmitchel | 2022-05-11 |
|      120 | tmitchel | 2022-05-09 |
|      121 | btang    | 2022-05-10 |
|      122 | yappiah  | 2022-05-11 |
|      123 | bmoreno  | 2022-05-10 |
|      124 | asundara | 2022-05-12 |
|      125 | bisles   | 2022-05-11 |
|      126 | jrafael  | 2022-05-12 |
|      127 | abellmas | 2022-05-09 |
|      128 | jclark   | 2022-05-09 |
|      129 | drosas   | 2022-05-12 |
|      130 | mrah     | 2022-05-11 |
|      131 | bisles   | 2022-05-09 |
|      132 | rjensen  | 2022-05-12 |
|      133 | asundara | 2022-05-12 |
|      134 | iuduike  | 2022-05-09 |
|      135 | bsand    | 2022-05-09 |
|      136 | mabadi   | 2022-05-10 |
|      137 | jrafael  | 2022-05-12 |
```

You have completed this activity and practiced applying

- the `WHERE` keyword
- the `BETWEEN` and `AND` operators, and
- operators for working with numeric or date and time data types (for example, `=`, `>`, `>=`)

to filter data from a table.

You're now ready to filter for numbers and dates to extract all sorts of useful data!

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.

Filters with AND, OR, and NOT

AND : both conditions must be met. First, `AND` is used to filter on two conditions. `AND` specifies that both conditions must be met simultaneously.

```
SELECT firstname, lastname, email, country, supportrepid
FROM customers
WHERE supportrepid = 5 AND country = 'USA';
```

OR : Either conditions can be met. The `OR` operator also connects two conditions, but `OR` specifies that either condition can be met. It returns results where the first condition, the second condition, or both are met.

```
SELECT firstname, lastname, email, country
FROM customers
WHERE country = 'Canada' OR country = 'USA';
```

Note: Even if both conditions are based on the same column, you need to write out both full conditions. For instance, the query in the previous example contains the filter `WHERE country = 'Canada' OR country = 'USA`

NOT: Unlike the previous two operators, the `NOT` operator only works on a single condition, and not on multiple ones. The `NOT` operator negates a condition. This means that SQL returns all records that don't match the condition specified in the query.

```
SELECT firstname, lastname, email, country

FROM customers
WHERE NOT country = 'USA';
```

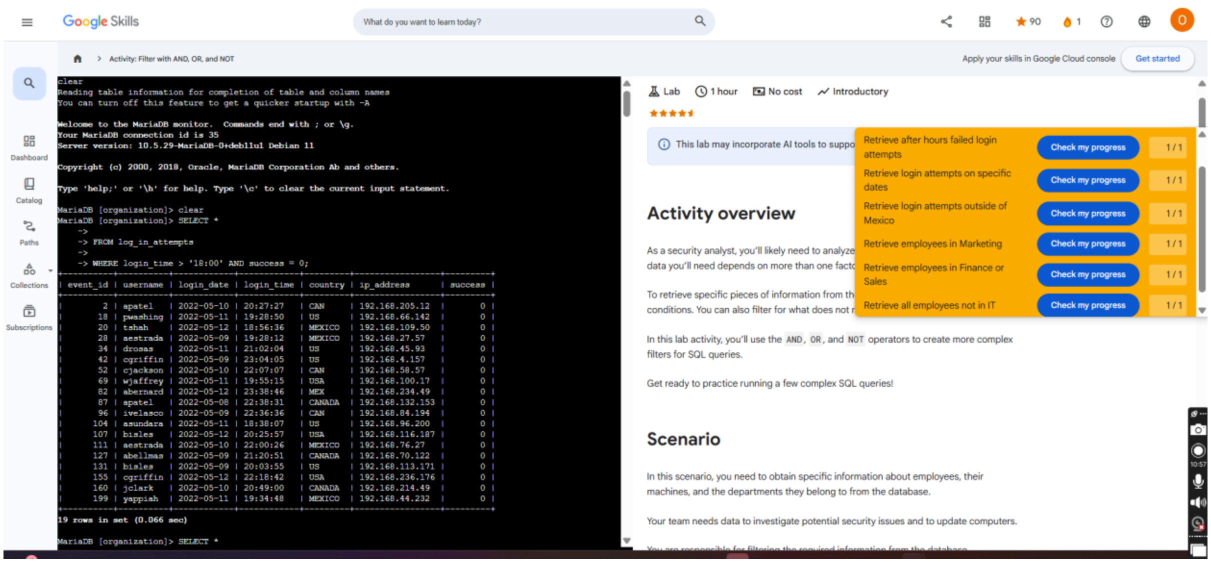
Combining logical operators

Logical operators can be combined in filters. For example, if you know that both the USA and Canada are not affected by a cybersecurity issue, you can combine operators to return customers in all countries besides these two. In the following query, `NOT` is placed before the first condition, it's joined to a second condition with `AND`, and then `NOT` is also placed before that second condition.

```
SELECT firstname, lastname, email, country

FROM customers
WHERE NOT country = 'Canada' AND NOT country = 'USA';
```

Practical Lab

```
-- Retrieve all failed login attempts that occurred after business hours
(after 18:00)

SELECT *

FROM log_in_attempts

WHERE login_time > '18:00' AND success = FALSE;


-- Retrieve all login attempts that occurred on May 8, 2022 or May 9, 2022

SELECT *

FROM log_in_attempts

WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';


-- Retrieve all login attempts that did not originate in Mexico

SELECT *

FROM log_in_attempts

WHERE NOT country LIKE 'MEX%';


-- Retrieve all employees in the Marketing department who are located in the
East building

SELECT *

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East%';


-- Retrieve all employees who work in either the Finance or Sales department

SELECT *

FROM employees

WHERE department = 'Finance' OR department = 'Sales';


-- Retrieve all employees who are not in the Information Technology department

SELECT *

FROM employees

WHERE NOT department = 'Information Technology';
```


Google Skills
What do you want to learn today?
90
1
Dashboard
Catalog
Paths
Collections
Subscriptions
Activity: Filter with AND, OR, and NOT
Apply your skills in Google Cloud console
Get started
Lab
1 hour
No cost
Introductory

```
clear
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 35
Server version: 10.5.29-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [organization]> clear
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142  |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50  |       0 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
|       52 | cjackson | 2022-05-10 | 22:07:07   | CAN     | 192.168.58.57   |       0 |
|       69 | wjaffrey | 2022-05-11 | 19:55:15   | USA     | 192.168.100.17  |       0 |
|       82 | abernard | 2022-05-12 | 23:38:46   | MEX     | 192.168.234.49  |       0 |
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
|       96 | ivelasco | 2022-05-09 | 22:36:36   | CAN     | 192.168.84.194  |       0 |
|      104 | asundara | 2022-05-11 | 18:38:07   | US      | 192.168.96.200  |       0 |
|      107 | bisles   | 2022-05-12 | 20:25:57   | USA     | 192.168.116.187 |       0 |
|      111 | aestrada | 2022-05-10 | 22:00:26   | MEXICO  | 192.168.76.27   |       0 |
|      127 | abellmas | 2022-05-09 | 21:20:51   | CANADA  | 192.168.70.122  |       0 |
|      131 | bisles   | 2022-05-09 | 20:03:55   | US      | 192.168.113.171 |       0 |
|      155 | cgriffin | 2022-05-12 | 22:18:42   | USA     | 192.168.236.176 |       0 |
|      160 | jclark   | 2022-05-10 | 20:49:00   | CANADA  | 192.168.214.49  |       0 |
|      199 | yappiah  | 2022-05-11 | 19:34:48   | MEXICO  | 192.168.44.232  |       0 |
+----------+----------+------------+------------+---------+-----------------+---------+
19 rows in set (0.066 sec)

MariaDB [organization]> SELECT *
```

This lab may incorporate AI tools to suppo

Retrieve after hours failed login attempts	Check my progress	1 / 1
Retrieve login attempts on specific dates	Check my progress	1 / 1
Retrieve login attempts outside of Mexico	Check my progress	1 / 1
Retrieve employees in Marketing	Check my progress	1 / 1
Retrieve employees in Finance or Sales	Check my progress	1 / 1
Retrieve all employees not in IT	Check my progress	1 / 1

Activity overview

As a security analyst, you'll likely need to analyze
data you'll need depends on more than one facto

To retrieve specific pieces of information from th
conditions. You can also filter for what does not r

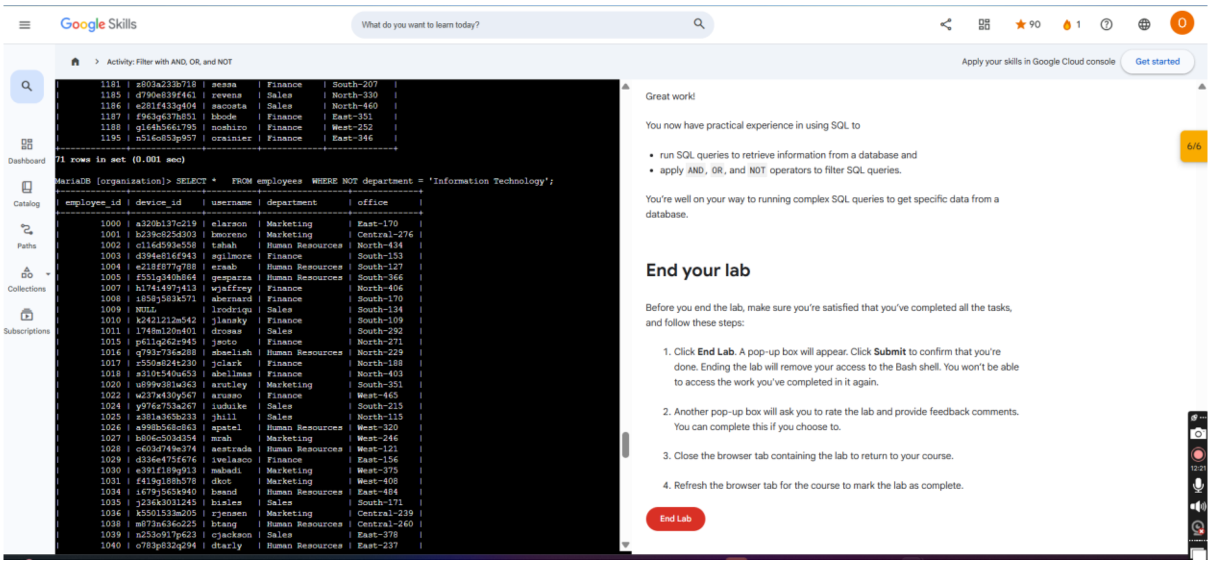
In this lab activity, you'll use the `AND`, `OR`, and `NOT` operators to create more complex filters for SQL queries.

Get ready to practice running a few complex SQL queries!

Scenario

In this scenario, you need to obtain specific information about employees, their machines, and the departments they belong to from the database.

Your team needs data to investigate potential security issues and to update computers.


Google Skills
What do you want to learn today?
90
1
Dashboard
Catalog
Paths
Collections
Subscriptions
Activity: Filter with AND, OR, and NOT
Apply your skills in Google Cloud console
Get started
6/6

```
|        1181 | z803a233b718 | sessa     | Finance         | South-207   |
|        1185 | d790e839f461 | revens    | Sales           | North-330   |
|        1186 | e281f433g404 | sacosta   | Sales           | North-460   |
|        1187 | f963g637h851 | bbode     | Finance         | East-351    |
|        1188 | g164h566i795 | noshiro   | Finance         | West-252    |
|        1195 | n516o853p957 | orainier  | Finance         | East-346    |
+-------------+--------------+-----------+-----------------+-------------+
71 rows in set (0.001 sec)

MariaDB [organization]> SELECT *   FROM employees  WHERE NOT department = 'Information Technology';
+-------------+--------------+-----------+-----------------+-------------+
| employee_id | device_id    | username  | department      | office      |
+-------------+--------------+-----------+-----------------+-------------+
|        1000 | a320b137c219 | elarson   | Marketing       | East-170    |
|        1001 | b239c825d303 | bmoreno   | Marketing       | Central-276 |
|        1002 | c116d593e558 | tshah     | Human Resources | North-434   |
|        1003 | d394e816f943 | sgilmore  | Finance         | South-153   |
|        1004 | e218f877g788 | eraab     | Human Resources | South-127   |
|        1005 | f551g340h864 | gesparza  | Human Resources | South-366   |
|        1007 | h174i497j413 | wjaffrey  | Finance         | North-406   |
|        1008 | i858j583k571 | abernard  | Finance         | South-170   |
|        1009 | NULL         | lrodriqu  | Sales           | South-134   |
|        1010 | k2421212m542 | jlansky   | Finance         | South-109   |
|        1011 | l748m120n401 | drosas    | Sales           | South-292   |
|        1015 | p611q262r945 | jsoto     | Finance         | North-271   |
|        1016 | q793r736s288 | sbaelish  | Human Resources | North-229   |
|        1017 | r550s824t230 | jclark    | Finance         | North-188   |
|        1018 | s310t540u653 | abellmas  | Finance         | North-403   |
|        1020 | u899v381w363 | arutley   | Marketing       | South-351   |
|        1022 | w237x430y567 | arusso    | Finance         | West-465    |
|        1024 | y976z753a267 | iuduike   | Sales           | South-215   |
|        1025 | z381a365b233 | jhill     | Sales           | North-115   |
|        1026 | a998b568c863 | apatel    | Human Resources | West-320    |
|        1027 | b806c503d354 | mrah      | Marketing       | West-246    |
|        1028 | c603d749e374 | aestrada  | Human Resources | West-121    |
|        1029 | d336e475f676 | ivelasco  | Finance         | East-156    |
|        1030 | e391f189g913 | mabadi    | Marketing       | West-375    |
|        1031 | f419g188h578 | dkot      | Marketing       | West-408    |
|        1034 | i679j565k940 | bsand     | Human Resources | East-484    |
|        1035 | j236k3031245 | bisles    | Sales           | South-171   |
|        1036 | k5501533m205 | rjensen   | Marketing       | Central-239 |
|        1038 | m873n636o225 | btang     | Human Resources | Central-260 |
|        1039 | n253o917p623 | cjackson  | Sales           | East-378    |
|        1040 | o783p832q294 | dtarly    | Human Resources | East-237    |
```

Great work!

You now have practical experience in using SQL to

- run SQL queries to retrieve information from a database and
- apply `AND`, `OR`, and `NOT` operators to filter SQL queries.

You're well on your way to running complex SQL queries to get specific data from a database.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.

End Lab

Joining in SQL

Code

```
-- Question 1: Retrieve all records from the machines table to review
available machine data.

SELECT *

FROM machines;
```

```
-- Question 2: Use an INNER JOIN to identify which employees are using which
machines

-- by matching records on the shared device_id column.

SELECT *

FROM machines

INNER JOIN employees

ON machines.device_id = employees.device_id;

-- Question 3: Use a LEFT JOIN to return all machines and any employees
assigned to them,

-- including machines that are not assigned to any employee.

SELECT *

FROM machines

LEFT JOIN employees

ON machines.device_id = employees.device_id;

-- Question 4: Use a RIGHT JOIN to return all employees and any machines
assigned to them,

-- including employees who do not have a machine assigned.

SELECT *

FROM machines

RIGHT JOIN employees

ON machines.device_id = employees.device_id;

-- Question 5: Use an INNER JOIN to retrieve all login attempts made by
employees

-- by joining the employees and log_in_attempts tables on the username column.

SELECT *

FROM employees

INNER JOIN log_in_attempts

ON employees.username = log_in_attempts.username;
```

Other functions in SQL

Aggregate functions

In SQL, **aggregate functions** are functions that perform a calculation over multiple data points and return the result of the calculation. The actual data is not returned.

There are various aggregate functions that perform different calculations:

- `COUNT` returns a single number that represents the number of rows returned from your query.
- `AVG` returns a single number that represents the average of the numerical data in a column.
- `SUM` returns a single number that represents the sum of the numerical data in a column.

Aggregate function syntax

To use an aggregate function, place the keyword for it after the `SELECT` keyword, and then in parentheses, indicate the column you want to perform the calculation on. Eg:

```
SELECT COUNT(firstname)
FROM customers;
```

If you want to find the number of customers from a specific country, you can add a filter to your query:

```
SELECT COUNT(firstname)
FROM customers
WHERE country = 'USA';
```