

# Apply filters to SQL queries

**Name:** Dhyanam

**Role:** Cyber Security Training

**Year:** 2026

## Project description:

- Scenario: As a Security Analyst I have to investigate recently discovered potential security issues that involve login attempts and employee machines. I have to examine the organization's data in their **employees** and **log\_in\_attempts** tables. I will also need to use SQL filters to retrieve records from different datasets and investigate the potential security issues

## Retrieve after hours failed login attempts:

- There was a potential security incident that occurred after business hours (after 18:00). All after hours login attempts that failed need to be investigated.
- The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.

```
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	

- The first part of the screenshot is my query, and the second part is a portion of the output. This query filters for failed login attempts that occurred after 18:00. First, I started by selecting all data from the **log\_in\_attempts** table. Then, I used a **WHERE** clause with an **AND** operator to filter my results to output only login attempts that occurred after 18:00 and were unsuccessful.

- The first condition is **login\_time > '18:00'**, which filters for the login attempts that occurred after **18:00**. The second condition is **success = '0'**, which filters for the failed login attempts.

## Retrieve login attempts on specific dates:

- A suspicious event occurred on 2022-05-09. Any login activity that happened on 2022-05-09 or on the day before needs to be investigated.
- The following code demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates.

```
MariaDB [organization]> select * from log_in_attempts where login_date = '2022-05-09' or login_date='2022-05-08';
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	

- The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all login attempts that occurred on **2022-05-09 or 2022-05-08**.
- First, I started by selecting all data from the **log\_in\_attempts** table. Then, I used a **WHERE** clause with an **OR** operator to filter my results to output only login attempts that occurred on either **2022-05-09 or 2022-05-08**. The first condition is **login\_date = '2022-05-09'**, which filters for logins on **2022-05-09**. The second condition is **login\_date = '2022-05-08'**, which filters for logins on 2022-05-08.

## Retrieve login attempts outside of Mexico:

- After investigating the organization's data on login attempts, I believe there is an issue with the login attempts that occurred outside of Mexico. These login attempts should be investigated.
- The following code demonstrates how I created a SQL query to filter for login attempts that occurred outside of Mexico.

```
MariaDB [organization]> select * from log_in_attempts where not country like 'Mex%';
```

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
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0
5	jrafael	2022-05-11	03:05:59	CANADA	192.168.86.232	1

- The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all login attempts that occurred in countries other than Mexico.
- First, I started by selecting all data from the **log\_in\_attempts** table. Then, I used a **WHERE** clause with **NOT** to filter for countries other than Mexico. I used **LIKE** with **MEX%** as the pattern to match because the dataset represents Mexico as **MEX** and **MEXICO**. The percentage sign (%) represents any number of unspecified characters when used with **LIKE**.

## Retrieve employees in Marketing:

- My team wants to update the computers for certain employees in the Marketing department.
- To do this, I have to get information on which employee machines to update.
- The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Marketing department in the East building.

```

MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE department='Marketing'
-> AND office LIKE 'East%';
+-----+-----+-----+-----+-----+
| employee_id | device_id | username | department | office |
+-----+-----+-----+-----+-----+
|          1000 | a320b137c219 | elarson | Marketing | East-170 |
|          1052 | a192b174c940 | jdarosa | Marketing | East-195 |
|          1075 | x573y883z772 | fbautist | Marketing | East-267 |
|          1088 | k865l965m233 | rgosh | Marketing | East-157 |
|          1103 | NULL | randerss | Marketing | East-460 |
|          1156 | a184b775c707 | dellery | Marketing | East-417 |
|          1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

```

- The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all employees in the Marketing department in the East building.
- First, I started by selecting all data from the employees table. Then, I used a **WHERE** clause with **AND** to filter for employees who work in the **Marketing** department and in the **East** building. I used **LIKE** with **East%** as the pattern to match because the data in the office column represents the East building with the specific office number. The first condition is the **department = 'Marketing'** portion, which filters for employees in the Marketing department. The second condition is the office **LIKE 'East%'** portion, which filters for employees in the East building.

## Retrieve employees in Finance or Sales:

- The machines for employees in the Finance and Sales departments also need to be updated.
- Since a different security update is needed, I have to get information on employees only from these two departments.
- The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales departments.

```
MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE department ='Finance' OR department='Sales';
```

employee_id	device_id	username	department	office
1003	d394e816f943	sgilmore	Finance	South-153
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170
1009	NULL	lrodriqu	Sales	South-134
1010	k242l212m542	jlansky	Finance	South-109
1011	l748m120n401	drosas	Sales	South-292

- The first part of the screenshot is my query, and the second part is a portion of the output. This query returns all employees in the Finance and Sales departments.
- First, I started by selecting all data from the employees table. Then, I used a **WHERE** clause with **OR** to filter for employees who are in the **Finance** and **Sales** departments. I used the **OR** operator instead of **AND** because I want all employees who are in either department. The first condition is **department = 'Finance'**, which filters for employees from the Finance department. The second condition is **department = 'Sales'**, which filters for employees from the Sales department.

## Retrieve all employees not in IT:

- My team needs to make one more security update on employees who are not in the Information Technology department. To make the update, I first have to get information on these employees.
- The following demonstrates how I created a SQL query to filter for employee machines from employees not in the Information Technology department.

```
MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE NOT department='Information Technology';
```

employee_id	device_id	username	department	office
1000	a320b137c219	el Larson	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

- The first part of the screenshot is my query, and the second part is a portion of the output. The query returns all employees not in the Information Technology department. First, I started by selecting all data from the employees table. Then, I used a **WHERE** clause with **NOT** to filter for employees not in this department.

## Summary:

- I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, log\_in\_attempts and employees. I used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.