

# Apply filters to SQL queries

## Project description

My organization is working to strengthen its system security. As part of my role, I ensure the system remains protected, investigate potential security issues, and update employee computers when necessary. Descriptions of how I utilized SQL with filters to carry out security-related tasks are given in the following sections.

## Retrieve after hours failed login attempts

A potential security incident occurred after business hours, after 18:00. Any failed login attempts made during this time need to be investigated.

Here is my code that shows 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 = FALSE;
```

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

The first part of my code shows my query, and the second part displays a portion of the output. This query identifies all failed login attempts that occurred after 18:00. I began by selecting all data from the log\_in\_attempts table. Then, I added a WHERE clause with an AND operator to filter the results to only those attempts made after 18:00 and marked as unsuccessful. The condition login\_time > '18:00' isolates attempts made after business hours, while success = FALSE filters for failed logins.

## Retrieve login attempts on specific dates

A suspicious event took place on 2022-05-09. Any login activity from that date or the day prior must be reviewed.

Here is my code that shows 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	0
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	0
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0

The first part of my code shows my query, and the second part displays a portion of the output. This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. I began by selecting all data from the log\_in\_attempts table. Then, I added a WHERE clause with an OR operator to filter the results to only those login attempts made on either 2022-05-09 or 2022-05-08. The first condition, login\_date = '2022-05-09', selects logins from 2022-05-09, while the second condition, login\_date = '2022-05-08', selects logins from the previous day.

## Retrieve login attempts outside of Mexico

After reviewing the organization's login attempt data, I noticed potential issues with attempts originating outside of Mexico. These login attempts should be investigated further..

Here is my code that shows 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	0
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	0

The first part of the code shows my query, and the second part displays a portion of the output. This query returns all login attempts that took place in countries other than Mexico. I began by selecting all data from the log\_in\_attempts table. Then, I added a WHERE clause using NOT to filter out any country values that represent Mexico. I used LIKE 'MEX%' as the pattern because the dataset records Mexico as both MEX and MEXICO. The percent symbol (%) allows the pattern to match any number of additional characters.

## Retrieve employees in Marketing

My team needs to update the computers for specific employees in the Marketing department, so I first need to gather information on which machines require updates.

Here is my code that shows 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 |
```

The first part of my code shows my query, and the second part displays a portion of the output. This query returns all employees in the Marketing department who work in the East building. I began by selecting all data from the employees table. Then, I used a WHERE clause with an AND operator to filter for employees who belong to the Marketing department and are located in the East building. I used LIKE 'East%' because the office column lists the East building along with specific office numbers. The first condition, department = 'Marketing', filters for Marketing employees, and the second condition, office LIKE 'East%', filters for those assigned to the East building.

## Retrieve employees in Finance or Sales

Additionally, the machines used by workers in the sales and finance divisions need to be updated. I only need to obtain personnel data from these two departments since a different security upgrade is required.

Here is my code that shows 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 |

```

The first part of the code shows my query, and the second part displays a portion of the output. This query returns all employees in the Finance and Sales departments. I began by selecting all data from the employees table. Then, I used a WHERE clause with an OR operator to filter for employees who belong to either department. I used OR instead of AND because I needed results from both groups. The first condition, department = 'Finance', filters for Finance employees, while the second condition, department = 'Sales', filters for Sales employees.

## Retrieve all employees not in IT

One more security update for staff members who are not in the IT department needs to be made by my team. I need to gather information on these employees before I can make the update.

Here is my code that shows 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 | elarson | Marketing | East-170 |
|          1001 | b239c825d303 | bmoreno | Marketing | Central-276 |
|          1002 | c116d593e558 | tshah | Human Resources | North-434 |

```

The first part of the code shows my query, and the second part displays a portion of the output. This query returns all employees who are not part of the Information Technology department. I began by selecting all data from the employees table, then used a WHERE clause with NOT to filter out employees in that department.

## Summary

I used filters in SQL queries to obtain particular data on employee machines and login attempts. I made use of two distinct tables: employees and log\_in\_attempts. I filtered for the precise data required for each task using the AND, OR, and NOT operators. Additionally, I filtered for patterns using LIKE and the percentage symbol (%) wildcard.