Apply filters to SQL queries

I'm a security professional at a company. Part of my job is to investigate security issues to help keep the system secure.
I've recently discovered some potential security issues that involve login attempts in employee machines. I'm tasked with examining the organization's data in their internal databases. For this task, I will use SQL queries to filter the data records from different datasets and investigate possible security issues.

Retrieve after hours failed login attempts

I recently discovered a potential security incident that occurred after business hours. To investigate this, I queried the `log_in_attempts` table and reviewed it for possible business after hours (`18:00`) login activity.

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_time > '18:00' AND success = FALSE;
```

The query selects all columns (`*`) from the table `log_in_attempts` in which there were login attempts after (`>`) `18:00`hs and they also failed to connect with the server (`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
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

This is the resulting table from the previous filtering query. We can observe that in the `success` column all of these login attempts failed (`0=FALSE`).
We can also see in the `login_time` column that all login attempts were done after `18:00`hs.
To connect these instructions I used the `AND` command.
From this table's data we can investigate the cause of the security incident that happened earlier.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate this event, I will review all login attempts which occurred on this day (2022-05-09) and the day before (2022-05-08).

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

The query selects all columns (*) from the table log_in_attempts in which there were login attempts on this day (2022-05-09) OR the day before (2022-05-08).

Retrieve login attempts outside of Mexico

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, I will need to investigate login attempts that occurred outside of Mexico.

```
MariaDB [organization]> SELECT *
    ->
    -> FROM log_in_attempts
    ->
    -> WHERE NOT country LIKE 'MEX%';
```

Using a similar query from the previous retrievals, in this case the WHERE line adds the NOT instruction so the resulting table doesn't have values that that contain words that start with the string 'MEX'(LIKE 'MEX%'). In this case I used the LIKE operator and not the = one because the log_in_attempts table has 'MEX' and 'MEXICO' as possible values for logins from Mexico in the country column.

Retrieve employees in Marketing

My team wants to perform security updates on specific employee machines in the Marketing department. I'm responsible for getting information on these employee machines and I will need to query the employees table.

```
MariaDB [organization]> SELECT *
    ->
    -> FROM employees
    ->
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
```

In this query we need the employees that are working in the Marketing department, so i typed department = 'Marketing'. Because we also need the employees that are simultaneously working in the east wing offices, I added the AND command for the additional instruction of searching all the offices in the office column (office LIKE 'East%').

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.

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

For this query I used a `WHERE` clause with `OR` to filter for employees who are in the Finance or Sales departments, which are both represented in the `department` column. The first condition is `department = 'Finance'`, which is responsible for filtering employees from the Finance department. The second condition is `department = 'Sales'`, which finds and filters all employees from the Sales Department. I used the `OR` connector because I need the information of the employees of either one of the departments.

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.

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

This case is very similar to the previous mexico case, where I used a `WHERE` clause with a `NOT` operand to get all the employees that do not belong to the Information Technology department represented in the `department` column.

Summary

Using filter queries I managed to get valuable information that was used thereafter to solve potential security issues in the company. To achieve this I used operators and SQL clauses to filter information from the company databases.