Apply filters to SQL queries

Project description

I am a security professional who works in a large organization. This task requires me to examine my organization's data in their log_in_attempts and employee SQL tables.

Retrieve after hours failed login attempts

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND
success = '0';
```

```
19 rows in set (0.087 sec)
```

A potential security incident occurred after business hours (post 18:00). All failed login attempts during these hours need to be investigated. This query focuses on failed login attempts after 18:00 with there being 19 attempts.

Retrieve login attempts on specific dates

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

```
75 rows in set (0.001 sec)
```

A suspicious event occurred on 2022-05-09. Any login activity on that date or the previous day needs to be investigated. This query returns all login attempts that occurred on 2022-05-09 or 2022-05-08. There were 75 attempts made on either of these days.

Retrieve login attempts outside of Mexico

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

```
144 rows in set (0.070 sec)
```

After investigating the organization's data on login attempts, I identified an issue with the login attempts that occurred outside of Mexico. This query returns all login attempts that occurred in countries other than Mexico and it came out to 144 attempts.

Retrieve employees in Marketing

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Marketing' AND of
fice 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 | k8651965m233 | 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 (0.029 sec)
```

My team needs to update the computers for certain employees in the Marketing department. To do this, I need to gather information on which employee machines to update. This query returns all employees in the Marketing department in the East building.

Retrieve employees in Finance or Sales

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Sales' OR departm
ent = 'Finance';
+-------------+--------------+----------+------------+-------------+
| 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 machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is required, I need to gather information on employees only from these two departments. This query returns all employees in the Finance and Sales departments.

Retrieve all employees not in IT

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

```
161 rows in set (0.001 sec)
```

My team needs to make one more security update for employees who are not in the Information Technology department. To make this update, I first need to gather information on these employees. This query returns all employees not in the Information Technology department and this turned out to be 161 employees

Summary

I applied filters to SQL queries to retrieve specific information on login attempts and employee machines. I utilized two of my organizations SQL tables, log_in_attempts and employees.