Project description

My organization is working to improve the security of its system. My role is to help keep the system secure, check any possible security problems, and update employee computers when necessary. The steps below show how I used SQL with filters to complete several security-related tasks.

Retrieve after hours failed login attempts

A possible security issue happened after business hours (after 18:00). Every failed login attempt during this time needs to be reviewed.

The code below shows how I created a SQL query to filter failed login attempts that took place 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 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
```

The first part of the screenshot shows my query, and the second part shows part of the output. This query filters all failed login attempts that happened after 18:00. I began by selecting all data from the **log_in_attempts** table. Then, I used a **WHERE** clause with an **AND** operator to return only the login attempts that happened after 18:00 and were unsuccessful. The first condition, **login_time > '18:00'**, filters attempts after 18:00. The second condition, **success = FALSE**, filters the failed attempts.

Retrieve login attempts on specific dates

A suspicious event took place on **2022-05-09**. Any login activity from that date or the day before needs to be checked.

The code below shows how I created a SQL query to filter login attempts on specific dates.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date = '2022-05-09' OR login_date = '2022-0
5-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|        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 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       25 | sbaelish | 2022-05-09 | 07:04:02   | US      | 192.168.33.137  |       1 |
```

The first part of the screenshot is my query, and the second part is part of the output. This query returns all login attempts from **2022-05-09** or **2022-05-08**. I started by selecting all data from the **log_in_attempts** table. Then, I used a **WHERE** clause with an **OR** operator to return only the attempts from these two dates. The first condition, **login_date = '2022-05-09'**, filters attempts on 2022-05-09. The second condition, **login_date = '2022-05-08'**, filters attempts on 2022-05-08.

Retrieve login attempts outside of Mexico

After reviewing the organization's login attempt data, I found a possible issue with attempts that came from outside Mexico. These attempts need further investigation.

The code below shows how I created a SQL query to filter 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  |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
```

The first part of the screenshot is my query, and the second part is part of the output. This query returns all login attempts that happened in countries other than Mexico. I began by selecting all data from the **log_in_attempts** table. Then, I used a **WHERE** clause with **NOT** to filter out Mexico. I used **LIKE** with **MEX%** because the dataset records Mexico as both **MEX** and **MEXICO**. The percentage sign **%** is a wildcard that represents any number of characters when used with **LIKE**.

Retrieve employees in Marketing

My team needs to update computers for some employees in the Marketing department. To do this, I must find which employee machines need updates.

The code below shows how I created a SQL query to filter employee machines for workers in the Marketing department located 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 | 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 (0.027 sec)
```

The first part of the screenshot is my query, and the second part is part of the output. This query returns all employees in Marketing who work in the East building. I started by selecting all data from the **employees** table. Then, I used a **WHERE** clause with **AND** to filter employees in the Marketing department and in the East building. I used **LIKE** with **East%** because the office numbers in the dataset start with "East."
The first condition, **department = 'Marketing'**, filters employees in Marketing.
The second condition, **office LIKE 'East%'**, filters employees in the East building.

Retrieve employees in Finance or Sales

Employees in the Finance and Sales departments also need an update, but they require a different security patch. I need to collect information on employees from these two departments only.

The code below shows how I created a SQL query to filter employee machines in Finance or Sales.

```
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 | k2421212m542 | jlansky  | Finance    | South-109   |
|        1011 | l748m120n401 | drosas   | Sales      | South-292   |
|        1015 | p611q262r945 | jsoto    | Finance    | North-271   |
```

The first part of the screenshot is my query, and the second part is part of the output. This query returns all employees in the Finance and Sales departments. I started by selecting all data from the **employees** table. Then, I used a **WHERE** clause with **OR** to include employees from both departments. I chose **OR** instead of **AND** because I want results from either department. The first condition, **department = 'Finance'**, filters Finance employees. The second condition, **department = 'Sales'**, filters Sales employees.

Retrieve all employees not in IT

My team needs to apply one more security update to employees who are not part of the Information Technology department. To do this, I first need to find these employees.

The code below shows how I created a SQL query to filter for 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   |
|        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   |
```

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

Summary

I used filters in SQL queries to find specific information about login attempts and employee machines. I worked with two tables: **log_in_attempts** and **employees**. I used the **AND**, **OR**, and **NOT** operators to filter data for each task. I also used **LIKE** and the **%** wildcard to filter pattern-based results.