Apply filters to SQL queries

Project description

The organization asked Cyber Security to investigate into security issues to help keep the system secure. SQL is an important tool in the world of cybersecurity and is essential when querying databases.The Cyber Security Team will examine the organization's data in their `employees` and `log_in_attempts` tables, the analyst will need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

Retrieve after hours failed login attempts

My Team was tasked to investigate how many failed login attempts happened after `'18:00'`

The following command was used on SQL to make the query to see who had an unsuccessful attempt, which I can identify with the `SUCCESS` column to show the number `0` only, which translates to `FALSE` (failed attempt). An additional command was given to order the result by login attempt time with `DESC`

```
SELECT *
-> FROM log_in_attempts
-> WHERE login_time > '18:00' AND success = 0
-> ORDER BY login_time DESC;
```

19 total failed attempts where made after 6pm. The string DESC (descending) was used for easier read/overview.

```
MariaDB [organization]> clear
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = 0
    -> ORDER BY login_time DESC;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|       82 | abernard | 2022-05-12 | 23:38:46   | MEX     | 192.168.234.49  |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
```

Retrieve login attempts on specific dates

All login attempts that occurred on 2022-05-09 and 2022-05-08 are listed based on `event_id`. The following string was used to pull the query

```
SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-08' OR login_date = '2022-05-09'
```

```
ORDER BY event_id;
```

The 0 and 1 in the SUCCESS column shows the value for failed attempts (0) and for success attempt (1)

75 total attempt Ire made in the shown time period 2022-05-08 till 2022-05-09

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

Retrieve login attempts outside of Mexico

Since the organization requested an investigation into suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico.

The following quire was used to exclude Mexico form the list, since Mexico is being entered differently into the database I used the `(%)` symbol after MEX to create a wildcard in combination with **`LIKE`** which is used with **`WHERE`** to search for a pattern in a column and to include every matching word after MEX.

```
SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
```

A total of 144 attempts Ire made 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

Retrieve employees in Marketing

The following syntax was used to identify employees in the East buildings `East-170,` `East-320,` and to separate them from `North-434`.

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

The First part of the syntax that I wrote says `SELECT *` which stands for SELECT and the Asterix says ALL. The second line says `FROM employees` stands for using the data from employees data table.

With `WHERE` I identify how I like to see the data. I used `LIKE` as a wildcard to tell the system that I want `'EAST%'` The `%` wildcard was used to identify buildings starting with East and to eliminate the North Building and its employees. `AND` is used to filter on two conditions.

AND specifies that both conditions must be met simultaneously.
With this query I identified 7 employees in the North Buildings that need their machines updated.

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

Retrieve employees in Finance or Sales
There is an update coming for the machines in Finance and Sales and the Organization has asked the Team to investigate two departments, employee categories, Sales and Finance. This was executed with the following syntax and I identified 71 employees with the following syntax. 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.

```
SELECT *
-> FROM employees
-> WHERE department = 'Finance' OR department = '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   |
```

Retrieve all employees not in IT

The following syntax was used to filter out people that are not in IT.
In this scenario below I see that employees where filtered out by using the **`NOT`** after **`WHERE`** which eliminated the employees containing the string **`'Information Technology'`** and displays the remaining data.

```
SELECT *
-> FROM employees
-> WHERE NOT department = 'Information Technology';
```

161 returned as a result.

```
ariaDB [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   |
```

Summary

I applied and managed to extract and separate requested data for the Organization for various operations by using SQL queries to get specific information on login attempts and employee machines. Two different tables where used such as `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.