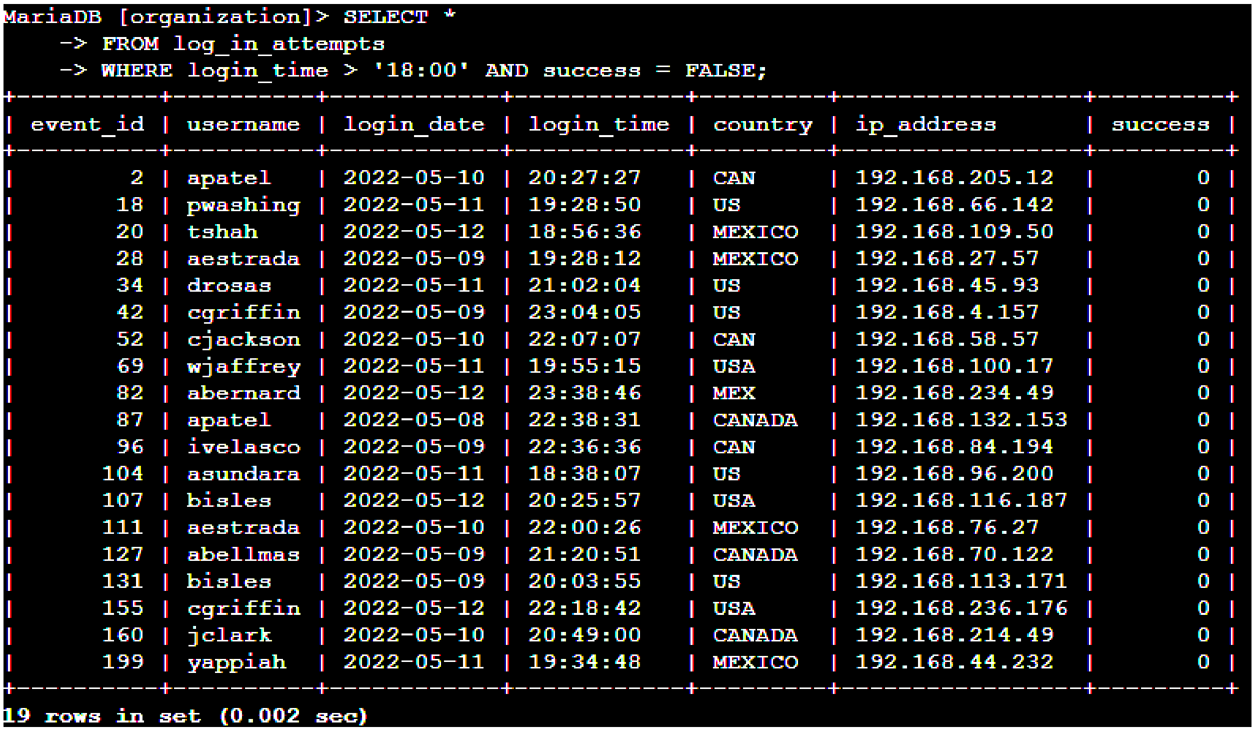
Apply filters to SQL queries

Project description

In this scenario I work with a mock database of employee records and login attempts at a fictitious large organisation. Recently, several security issues have been discovered and login records need to be reviewed to ensure ongoing security.

Retrieve after hours failed login attempts

Using SQL filtering, a list of unsuccessful login attempts made outside of business hours (finishing at 6pm in this organisation) can be retrieved.

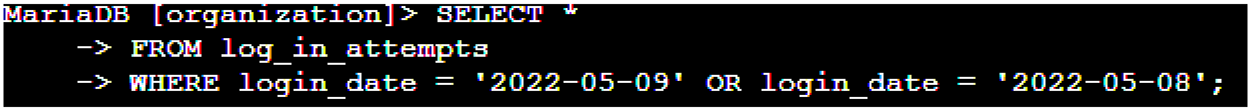


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
|       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 |
+----------+----------+------------+------------+---------+-----------------+---------+
19 rows in set (0.002 sec)
```

We can see that 19 such login attempts were made.

Retrieve login attempts on specific dates

A suspicious event occurred on May 9th 2022. Login attempts that occurred on this day or the day before need to be reviewed. This can be done with an SQL query using the OR operator.

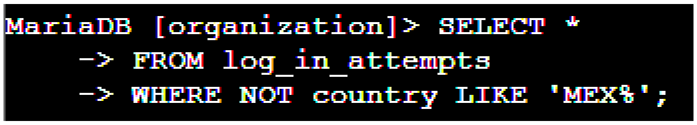


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

This query returns 75 records from the database.

Retrieve login attempts outside of Mexico

We need to review login attempts that were made from outside Mexico. This can be done with a query using the NOT operator.

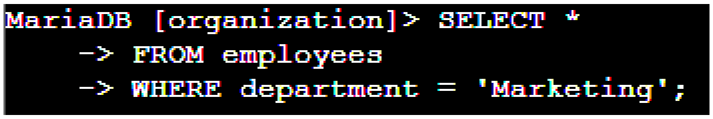


```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
```

This query returns 144 records of login attempts made outside of Mexico.

Retrieve employees in Marketing

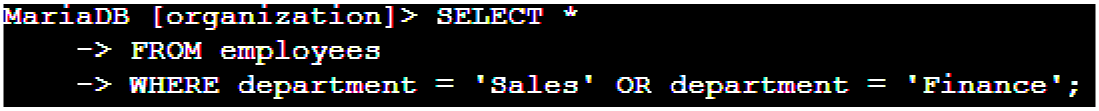
We now need to retrieve a list of employees in the Marketing department. This is done using SQL filtering, returning 44 records from the database.



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

Retrieve employees in Finance or Sales

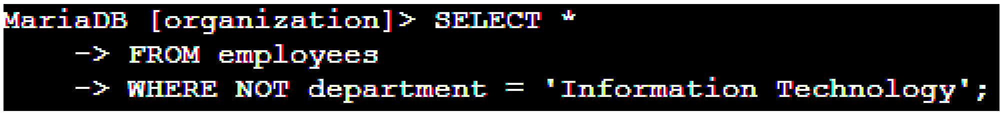
We also need a list of employees who work in either Finance or Sales. This can be done using the OR operator. This query returns 71 records.



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

Retrieve all employees not in IT

We can retrieve a list of employees who do not work in the IT department using the NOT operator. This returns 161 records.



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

Summary

Using SQL, we have examined various relevant login attempts and employee records to identify any potential security breaches.