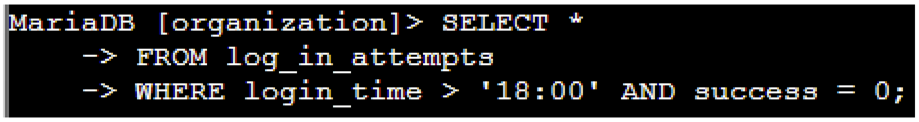
Apply filters to SQL queries

Project description

In this project, I use SQL to search for, filter, and sort through data for my organization to help keep our system secure. I go through the date, time, and locations of login attempts, and sort employees by various departments.

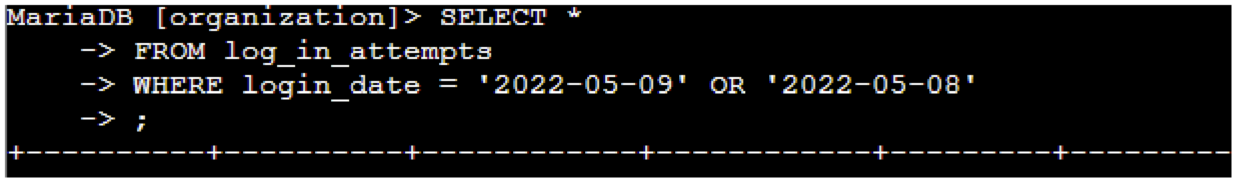
Retrieve after-hours failed login attempts



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

In order to identify failed login attempts after 18:00, I used this SQL Query. The `SELECT *` grabs all columns and the `FROM log_in_attempts` tells my command to grab the columns from the `log_in_attempts` table. I then use the line: `WHERE login_time > ‘18:00’ AND success = 0;` Here, the where gives the condition of only selecting the item if the login time exceeds 18:00 and is a failed login attempt. The `AND` is exclusive, meaning both conditions must be satisfied.

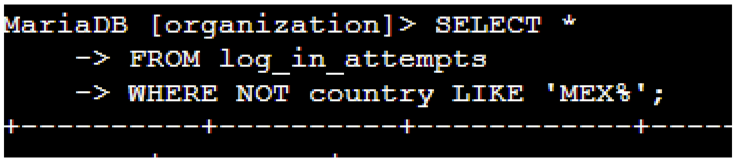
Retrieve login attempts on specific dates



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

I used the `WHERE` statement to investigate only the logins on the dates `2022-05-09` and `2022-05-08`. Just like the previous example, the `SELECT *` selects every column and the from tells the command to take the columns from the `log_in_attempts table`. The `WHERE` is a condition that displays the login attempt only if it was made on either of those two dates. While the previous `AND` was exclusive, this `OR` is inclusive, meaning that the log in attempt could fall on either one of the dates.

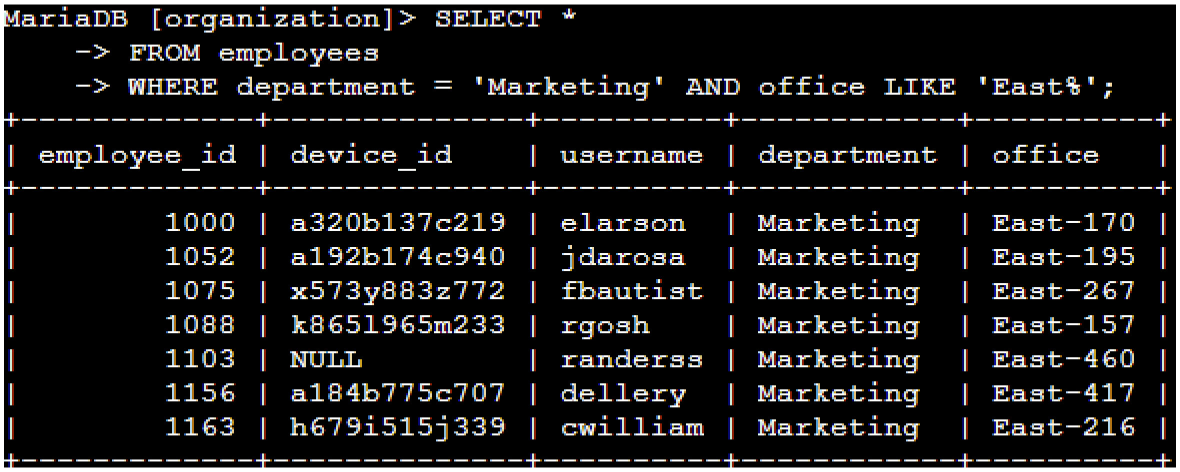
Retrieve login attempts outside of Mexico



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

Then, I needed to investigate login attempts that originated from outside of Mexico and I used the NOT and LIKE keywords to help me achieve this. After selecting the table, I specified `WHERE NOT`, which means where the condition does not match the following. I then typed country `LIKE 'MEX%';` to look for strings in the country column beginning with `MEX`.

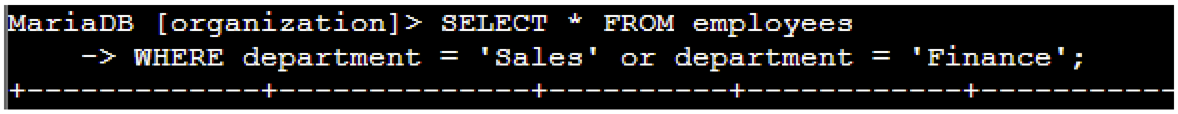
Retrieve employees in Marketing



```
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 |
+-------------+---------------+----------+------------+----------+
```

My next objective was to identify employees in the marketing department who worked in the East building as they were due for security updates. I selected all columns with `SELECT *` and used `FROM employees` so that the command grabbed the employees table. I then specified the department with the equals to operator: `department = 'Marketing'`. Finally, I typed `AND office LIKE 'East%';` to exclusively select employees from the Marketing department that worked in the East building. The `'%'` operator specifies any number of succeeding characters.

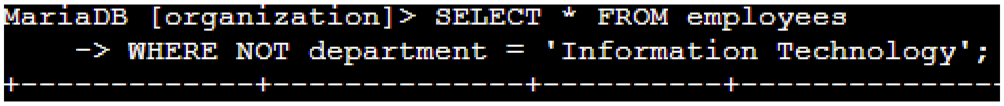
Retrieve employees in Finance or Sales `%`



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

I then had to identify the employee machines from the Sales or Finance departments as they needed a different security update. `SELECT * FROM employees` selects all columns from the employees table. Next, `WHERE department = ‘Sales’ or Department = ‘Finance’;` Allows me to specify either the Sales or Finance department and print out both.

Retrieve all employees not in IT



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

Finally, I had to identify all the departments besides the Information Technology department as they had already gotten the update that all the other departments are still due for. I used `WHERE NOT department = ‘Information Technology’` to select only departments that weren’t Information Technology.

Summary

In summary, I have used SQL to filter through log in attempts, using specific dates, locations and times. I have also filtered through different departments in order to deploy needed patch updates. I have used several different keywords within SQL including AND, OR, NOT, WHERE, LIKE, SELECT, and FROM.