Filter with AND, OR and NOT

Skills Acquired:

- Run SQL queries to retrieve information from a database
- Apply AND, OR, and NOT operators to filter SQL queries

Scenario:

In this scenario, you need to obtain specific information about employees, their machines, and the departments they belong to from the database.

Your team needs data to investigate potential security issues and to update computers.

You are responsible for filtering the required information from the database.

Task 1. Retrieve after hours failed login attempts

Your team is investigating failed login attempts that were made after business hours. You want to retrieve this information from the login activity. You'll identify all unsuccessful attempts after 18:00.

The login_time column in the log_in_attempts table contains information on when login attempts were made. Office hours end at '18:00'.

The success column in the log_in_attempts table contains values of TRUE or FALSE to indicate whether the login was successful. MySQL stores Boolean values as 1 for TRUE, and 0 for FALSE. This means that TRUE is represented as 1, and FALSE represented as 0 in the success column.

- Use the AND operator to retrieve the failed login attempts that occurred after business hours. Replace the X and Y with the correct values to filter for the records you need:

```
SELECT *
FROM log_in_attempts
WHERE login_time > 'X' AND success = Y;
```

- **How many failed login attempts occurred after 18:00?**

  19

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

```
+----------+----------+------------+------------+---------+----------------+
---------+
19 rows in set (0.001 sec)
```

Task 2. Retrieve login attempts on specific dates

Your team is investigating a suspicious event that occurred on '2022-05-09'. You want to retrieve all login attempts that occurred on this day and the day before ('2022-05-08').

The login_date column in the log_in_attempts table contains information on the dates when login attempts were made.

- Use the OR operator to retrieve the failed login attempts on the specified days. Replace the X and Y with the correct values to filter for the records you need:

```
SELECT *
FROM log_in_attempts
WHERE login_date = 'X' OR login_date = 'Y';
```

- **How many login attempts were made on these two days?**

  75

Task 3. Retrieve login attempts outside of Mexico

Now, your team is investigating logins that did not originate in Mexico, and you need to find this information. Note that the country field includes entries with 'MEX' and 'MEXICO'. You should use the NOT and LIKE operators and the matching pattern 'MEX%'.

- Run the following SQL query to retrieve login attempts that did not originate in Mexico. Replace X with the correct operator and Y with the correct pattern to filter for the information you need:

```
SELECT *
FROM log_in_attempts
WHERE X country LIKE 'Y';
```

- **How many login attempts were made outside of Mexico?**

  144

Task 4. Retrieve employees in Marketing

For tasks 4, 5 and 6 you need to retrieve the information from the department and office columns in the employees table.

You can run the following SQL query if you need to view the columns and values in the employees table:

```
SELECT *
FROM employees;
```

Your team is updating employee machines, and you need to obtain the information about employees in the 'Marketing' department who are located in all offices in the East building (such as 'East-170' or 'East-320').

- Write a SQL query to retrieve this information from the employees table. Select all columns and include filters on the department and office columns to return only the needed records.
- **What is the username of the first employee in the Marketing department in the East building?**

  Elarson

```
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 (0.001 sec)
```

Task 5. Retrieve employees in Finance or Sales

Now, your team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and you need to locate information on these employees.

- Write a SQL query to retrieve records for employees in the 'Finance' or the 'Sales' department.
- **What is the username of the first employee in the Sales department returned by the query?**

  Lrodriqu

```
MariaDB [organization]> select *
    -> from employees
    -> where department = 'Sales';
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
+-------------+-------------+----------+------------+-------------+
|        1009 | NULL        | lrodriqu | Sales      | South-134   |
```

Task 6. Retrieve all employees not in IT

Your team needs to make one more update. This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department. You should use the NOT operator to identify these employees.

- Write a SQL query to retrieve records for employees who are not in the 'Information Technology' department.

- **How many employees are not in the Information Technology department?**

  161

```
MariaDB [organization]> select *
    -> from employees
    -> where not department = 'Information Technology';
```

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