Apply filters to SQL queries

Project description

1. You recently discovered a potential security incident that occurred after business hours. To investigate this, you need to query the **log_in_attempts** table and review **after hours login** activity. Use filters in SQL to create a query that identifies **all failed login attempts that occurred after 18:00.** (The time of the login attempt is found in the **login_time** column. The **success** column contains a **value of 0** when a login attempt **failed**; you can use either a value of 0 or FALSE in your query to identify failed login attempts.)
2. A suspicious event occurred on **2022-05-09**. To investigate this event, you want to review all login attempts which **occurred on this day and the day before**. Use filters in SQL to create a query that identifies all login attempts that occurred on **2022-05-09 or 2022-05-08.** (The date of the login attempt is found in the **login_date** column.)
3. There's been suspicious activity with login attempts, but the team has determined that this activity **didn't originate in Mexico**. Now, you need to investigate **login attempts** that occurred **outside of Mexico**. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico. (When referring to Mexico, the **country** column contains values of both **MEX and MEXICO**.)
4. Your team wants to perform security updates on specific **employee** machines in the **Marketing** department. You're responsible for getting information on these employee machines and will need to query the employees table. Use filters in SQL to create a query that **identifies** all **employees** in the **Marketing** department for **all offices** in the **East** building.

(The department of the employee is found in the **department** column, which contains values that include Marketing. The office is found in the **office** column. Some examples of values in this column are East-170, East-320, and North-434.)

5. Your team now needs to perform a different security update on machines for employees in the **Sales** and **Finance** departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments. (The department of the employee is found in the **department** column, which contains values that include Sales and Finance.)

6. Your team needs to make one more update to employee machines. The employees who are in the Information Technology department already had this update, but employees in all other departments need it. Use filters in SQL to create a query which identifies all employees not in the IT department. (The department of the employee is found in the department column, which contains values that include Information Technology.)

Describe your query and how it works.

Retrieve after hours failed login attempts

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

`Select` follows the fields you want to display in the output

`*` Is a wildcard character used to indicate all fields will be displayed in output

`From` follows with the name of table from which we want to display data

`Where` allows us to search the data and filter it using criteria mentioned after the keyword

Since we need after hours login attempts (`Login_time > '18:00'`) and failed login attempts (`success = 0`), we use the conditional operator `AND` between the two conditions.

Retrieve login attempts on specific dates

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

We need to find the records of login attempts made on 2022-05-09 or 2022-05-08. This is the reason we use the conditional operator `OR` between the two conditions. The SQL syntax doesn't permit us to use `OR` between two dates; rather, we need to specify the complete statements (`login_date = '2022-05-09'` and `login_date ='2022-05-08'`) and then separate the two using the `OR` operator.

Retrieve login attempts outside of Mexico

```
SELECT *
FROM log_in_attempts
WHERE NOT country LIKE 'Mex%';
```

`NOT` keyword is used for negation. The output of a conditional statement is negated and then applied.

When we say `WHERE NOT country = 'Mexico'`, it means search records where the country is Not Mexico.

`LIKE` keyword is used when we need to use special characters instead of finding exactly equal strings. For example, when we say `WHERE country LIKE 'Mex%'` it matches any strings beginning with 'Mex'. It would match mex, mexican, mexico etc.

When we use the `NOT` operator before the statement, it means to select all countries which are not 'Mex' or beginning with the string 'Mex'.

Retrieve employees in Marketing

```
SELECT *
FROM employees
WHERE department = 'Marketing' AND office LIKE 'EAST%';
```

This query searches and filters the data in employees table, and displays only those employee records who are in the Marketing department and in the East building of organizations offices.

Retrieve employees in Finance or Sales

```
SELECT *
FROM employees
WHERE department = 'Finance' OR department = 'Sales;
```

This query returns records of employees who are in Finance or Sales department.

Retrieve all employees not in IT

These are multiple ways of getting the same results, that is finding record of employees who are not in IT department.

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

OR

```
SELECT *
FROM employees
WHERE department != 'Information Technology';
```

OR

```
SELECT *
FROM employees
WHERE department <> 'Information Technology';
```

Summary

SQL - Structured Query Language is the language of databases. It allows searching, creating, deleting records in the database. It is the way we can communicate with databases. Databases play a vital role in organizing and storing organisations data. This is the reason as security analysts we need to know SQL to be able to query and find relevant records related to security.