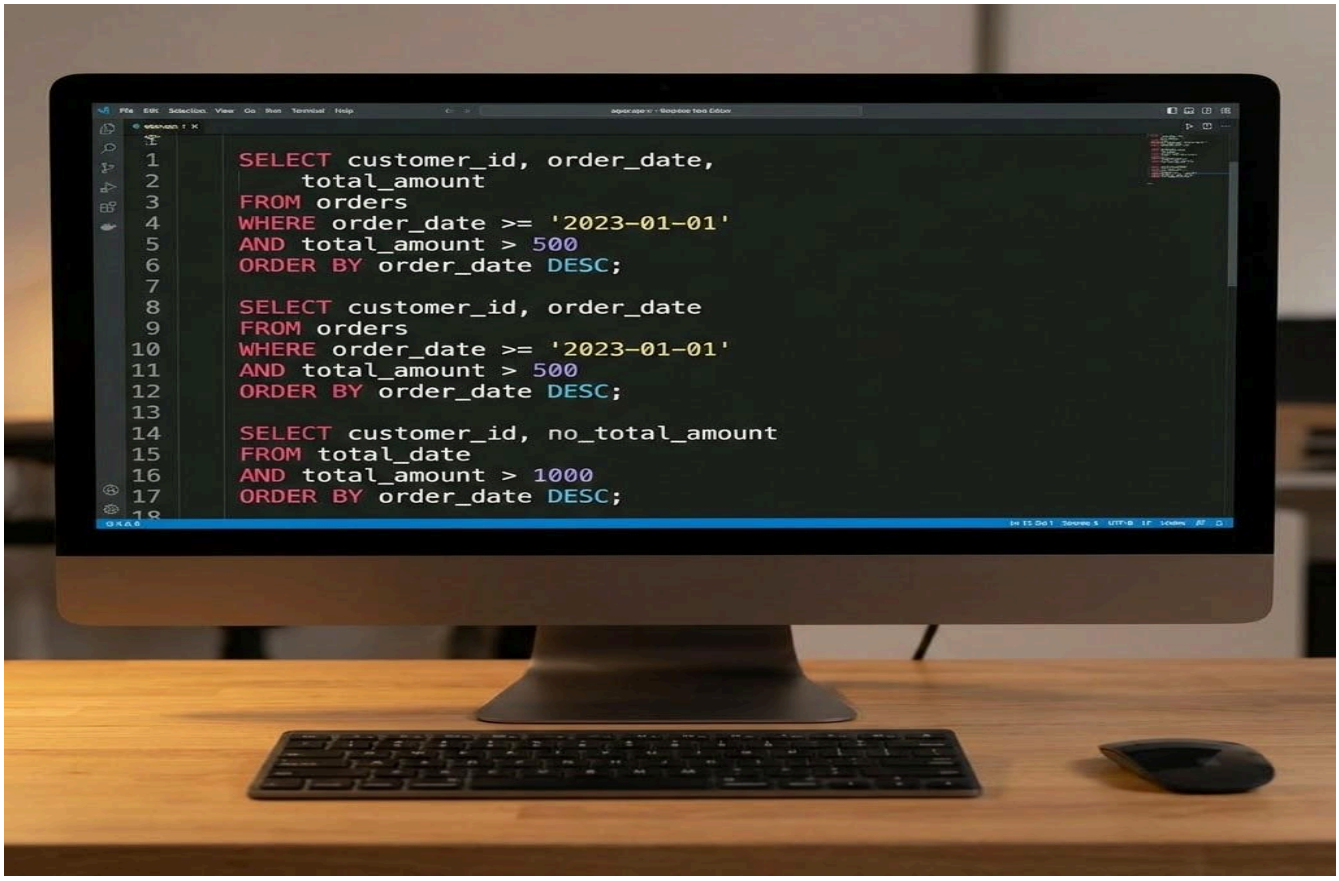

Apply filters to SQL queries Overview

Scenario

You are a security professional at a large organization tasked with investigating potential security issues involving login attempts and employee machines. You will examine the organization's data in the `employees` and `log_in_attempts` tables, using SQL filters to retrieve specific records and investigate the potential security issues.

SQL Query Demonstrations

Retrieve after hours failed login attempts

Goal: Identify all failed login attempts that occurred after 18:00. This uses the `AND` operator to combine two filtering conditions.

Query Description:
The query selects all columns (`*`) from the `log_in_attempts` table. It uses the `WHERE` clause to apply two conditions joined by the `AND` operator:

1. `success = 0` (or `success = FALSE`) to filter for failed login attempts.
2. `login_time > '18:00:00'` to filter for attempts that occurred after 6:00 PM.

SQL Query:SELECT

*

FROM

log_in_attempts

WHERE

success = 0

AND login_time > '18:00:00';

Retrieve login attempts on specific dates

Goal: Identify all login attempts that occurred on 2022-05-09 or 2022-05-08. This uses the `OR` operator to check for multiple values in a single column.

Query Description:
The query selects all columns (`*`) from the `log_in_attempts` table. It uses the `WHERE` clause and the `OR` operator to filter records where the `login_date` column is either equal to `'2022-05-09'` or equal to `'2022-05-08'`.

SQL Query:SELECT

*

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

Retrieve login attempts outside of Mexico

Goal: Identify all login attempts that originated outside of Mexico. This uses the `NOT` operator in conjunction with `LIKE` to exclude specific country names.

Query Description:
The query selects all columns (`*`) from the `log_in_attempts` table. It uses the `WHERE` clause with the `NOT` operator and the `LIKE` keyword to exclude any country value that contains the string `'MEX'`. This handles both 'MEX' and 'MEXICO' due to the wildcard character `%`.

SQL Query:

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

Retrieve employees in Marketing (East Building)

Goal: Identify all employees in the Marketing department who work in an office located in the East building. This query uses the `AND` operator and the `LIKE` keyword.

Query Description:
The query selects all columns (`*`) from the `employees` table. It uses the `WHERE` clause with the `AND` operator to combine two conditions:

1. `department LIKE '%Marketing%'` to filter for employees in the Marketing department (using `%` to match variations like 'Marketing').

2. `office LIKE 'East-%'` to filter for offices located in the East building (e.g., 'East-170', 'East-320').

SQL Query:SELECT

*

FROM

employees

WHERE

department LIKE '%Marketing%'

AND office LIKE 'East-%';

Retrieve employees in Finance or Sales

Goal: Identify all employees who belong to either the Sales or Finance departments. This uses the `OR` operator to include records matching one of two possible department values.

Query Description:
The query selects all columns (`*`) from the `employees` table. It uses the `WHERE` clause with the `OR` operator to filter records where the `department` column contains either the string `'Sales'` or the string `'Finance'`.

SQL Query:SELECT

*

FROM

employees

WHERE

department LIKE '%Sales%'

OR department LIKE '%Finance%';

Retrieve all employees not in IT

Goal: Identify all employees who are not in the Information Technology (IT) department. This uses the `NOT` operator and `LIKE` to exclude a specific department.

Query Description:
The query selects all columns (`*`) from the `employees` table. It uses the `WHERE` clause with the `NOT` operator and the `LIKE` keyword to exclude any employee whose `department` column contains the string `'Information Technology'`.

SQL Query:SELECT

*

FROM

employees

WHERE

department NOT LIKE '%Information Technology%';