Apply Filters to SQL Queries

Project description

In this project, I used SQL queries to extract specific information from organizational databases by applying logical filters. This included retrieving filtered login attempts and employee data by using operators such as `AND`, `OR`, `NOT`, and pattern matching with `LIKE`, as well as filtering by dates and times. These skills help in narrowing down data sets for security audits, troubleshooting, and updating employee systems.

Retrieve after hours failed login attempts

I retrieved all login attempts that failed and occurred after normal business hours (after 18:00). I used the `AND` operator to combine two conditions: `login_time > '18:00'` and `success = 0`, where 0 represents a failed login in the Boolean column. This type of time filtering helps identify potential unauthorized access outside approved working hours.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time > '18:00.00' A
ND success = '0';
```

Retrieve login attempts on specific dates

To investigate a suspicious event, I retrieved all login attempts that occurred on May 8 and May 9, 2022. I used the `OR` operator to include both dates in the query: `login_date = '2022-05-08' OR login_date = '2022-05-09'`. Filtering by specific dates helps isolate events and understand access patterns around known incidents.

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

Retrieve login attempts outside of Mexico

I retrieved login attempts that did not originate in Mexico. Since the country field had variations like 'MEX' and 'MEXICO', I used the `LIKE` operator with a wildcard and combined it with `NOT` to exclude both: `NOT country LIKE 'MEX%'`. The `%` wildcard matched any characters following 'MEX', ensuring both abbreviations and full names were excluded.

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

Retrieve employees in Marketing

I retrieved records of employees working in the Marketing department and located in East building offices, such as East-170. I combined filters with the `AND` operator and used `LIKE 'East%'` to match all office names beginning with "East". The `LIKE` filter is useful for flexible matching when exact values may vary but follow a consistent pattern.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Marketing' AND of
fice LIKE 'East%';
```

Retrieve employees in Finance or Sales

To prepare for system updates, I identified employees in either the Finance or Sales departments using the `OR` operator in the query: `department = 'Finance' OR department = 'Sales'`. The `OR` filter is helpful when you need to include multiple values for the same column in your results.

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

Retrieve all employees not in IT

I retrieved employee records excluding anyone in the Information Technology department by using the `NOT` filter: `NOT department = 'Information Technology'`. The `NOT` operator is critical for excluding specific values when you only want to include the remaining set of data.

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

Summary

In this project, I learned how to apply logical filters in SQL to return precise and relevant data. I practiced using `AND` for combining multiple conditions that must all be true, `OR` to include alternative matching values, `NOT` to exclude certain records, and `LIKE` for flexible pattern matching with wildcards. I also worked with date and time filters to narrow down login activity based on specific timeframes. These techniques are essential for managing large datasets, improving system security, and performing targeted system maintenance across an organization.