Report Prepared: Roopa Patel
Dated: 14th February 2024.

Report - Apply filters to SQL queries

Project description

In my role as a security professional, I leverage SQL to meticulously investigate security issues and potential vulnerabilities within our organization's systems. Through targeted queries into our databases, notably the employees and log_in_attempts tables, I adeptly retrieve pertinent information to discern security incidents, track login attempts, and delve into potential breaches. Employing SQL filters, including the AND, OR, and NOT operators, I methodically filter and analyze data, unveiling specific patterns or anomalies that may signify security threats.

My work in SQL plays a pivotal role in bolstering our organization's cybersecurity efforts. By proactively identifying and addressing security risks, I ensure the integrity and confidentiality of our data while safeguarding the entire computing environment for employees and stakeholders alike. To provide comprehensive insight into our database schema and table layouts, I've compiled a Table formats document, detailing the structural intricacies of each table. You can access this document , enriching your understanding of our data architecture and aiding in your review of my portfolio.

Retrieve after hours failed login attempts

In this task, we retrieved login attempts that occurred on specific dates, '2022-05-09' and '2022-05-08', to investigate a suspicious event. The query used the `OR` operator to combine conditions for both dates. Here's the SQL query and explanation:

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

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+----------------+---------+
| event_id | username | login_date | login_time | country | ip_address     | success |
+----------+----------+------------+------------+---------+----------------+---------+
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12 |       0 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142 |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50 |       0 |
```

Explanation:

- `SELECT *`: Selects all columns from the log_in_attempts table.
- `FROM log_in_attempts`: Specifies the table from which to retrieve data.
- `WHERE login_date = '2022-05-09' OR login_date = '2022-05-08'`:
- Filters the records to include only login attempts that occurred on '2022-05-09' or '2022-05-08'. The `OR` operator is used to combine multiple conditions, allowing us to retrieve data for both specified dates simultaneously.

Retrieve login attempts on specific dates

In this task, we retrieved login attempts that did not originate in Mexico by using the `NOT` operator along with the `LIKE` operator and the pattern 'MEX%'. Here's the SQL query and explanation:

1. SELECT *
2. FROM log_in_attempts
3. WHERE NOT country LIKE 'MEX%';

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+----------------+---------+
| event_id | username | login_date | login_time | country | ip_address     | success |
+----------+----------+------------+------------+---------+----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

Explanation:

- `SELECT *`: Selects all columns from the log_in_attempts table.
- `FROM log_in_attempts`: Specifies the table from which to retrieve data.
- `WHERE NOT country LIKE 'MEX%'`:

- Filters the records where the country does not start with 'MEX'. The `LIKE` operator is used to match patterns, and 'MEX%' matches any country name that starts with 'MEX'. By using `NOT`, we retrieve records where the country does not match this pattern, effectively excluding logins from Mexico.

Retrieve login attempts outside of Mexico

In this task, we retrieved login attempts that did not originate in Mexico by using the `NOT` operator along with the `LIKE` operator and the pattern 'MEX%'. Here's a breakdown of the SQL query used:

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

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       0 |
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
```

Explanation:

- `SELECT *`: Selects all columns from the log_in_attempts table.
- `FROM log_in_attempts`: Specifies the table from which to retrieve data.
- `WHERE NOT country LIKE 'MEX%'`:
- Filters the records where the country does not start with 'MEX'. The `LIKE` operator is used to match patterns, and 'MEX%' matches any country name that starts with 'MEX'. By using `NOT`, we retrieve records where the country does not match this pattern, effectively excluding logins from Mexico.

Retrieve employees in Marketing

To retrieve employees in the Marketing department who are located in all offices in the East building, we can use the following SQL query:

1. SELECT *
2. FROM employees
3. WHERE department = 'Marketing' AND office LIKE 'East-%';

```
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 |
```

Explanation:

- `SELECT *`: Selects all columns from the employees table.
- `FROM employees`: Specifies the table from which to retrieve data.
- `WHERE department = 'Marketing'`: Filters the records to include only employees in the Marketing department.
- `AND office LIKE 'East-%'`:
- Further filters the records to include only employees whose office starts with 'East-'. The `%` wildcard character is used to match any sequence of characters following 'East-'.

Retrieve employees in Finance or Sales

To retrieve records for employees in the Finance or Sales department, we can use the following SQL query:

1. SELECT *
2. FROM employees
3. WHERE department = 'Finance' OR department = 'Sales';

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+--------------+----------+------------+-------------+
| employee_id | device_id    | username | department | office      |
+-------------+--------------+----------+------------+-------------+
|        1003 | d394e816f943 | sgilmore | Finance    | South-153   |
|        1007 | h174i497j413 | wjaffrey | Finance    | North-406   |
|        1008 | i858j583k571 | abernard | Finance    | South-170   |
```

Explanation:

- `SELECT *`: Selects all columns from the employees table.
- `FROM employees`: Specifies the table from which to retrieve data.
- `WHERE department = 'Finance' OR department = 'Sales'`:
- Filters the records to include only employees in either the Finance or Sales department. The `OR` operator is used to specify that records meeting either condition are included.

Retrieve all employees not in IT

To retrieve records for employees who are not in the Information Technology (IT) department, we can use the following SQL query:

1. SELECT *
2. FROM employees
3. WHERE NOT department = 'Information Technology';

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department = 'Information Technology';
+-------------+--------------+----------+-----------------+-------------+
| employee_id | device_id    | username | department      | office      |
+-------------+--------------+----------+-----------------+-------------+
|        1000 | a320b137c219 | elarson  | Marketing       | East-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing       | Central-276 |
|        1002 | c116d593e558 | tshah    | Human Resources | North-434   |
```

Explanation:

- `SELECT *`: Selects all columns from the employees table.
- `FROM employees`: Specifies the table from which to retrieve data.
- `WHERE NOT department = 'Information Technology'`:
- Filters the records to include only employees whose department is not Information Technology. The `NOT` operator negates the condition, selecting records that do not match the specified criteria.

Summary

Retrieve after hours failed login attempts: The SQL query filtered login attempts that occurred after business hours and were unsuccessful. This task demonstrated the use of the `AND` operator to combine conditions.

Retrieve login attempts on specific dates: The SQL query retrieved login attempts that occurred on specific dates. It utilized the `OR` operator to filter records based on multiple date conditions.

Retrieve login attempts outside of Mexico: This task involved retrieving login attempts originating from countries other than Mexico. The SQL query used the `NOT` operator in conjunction with the `LIKE` operator to exclude records with country codes starting with 'MEX' or 'MEXICO'.

Retrieve employees in Marketing: A SQL query was used to retrieve information about employees in the Marketing department located in offices within the East building. This task showcased filtering based on multiple conditions using the `AND` operator.

Retrieve employees in Finance or Sales: The SQL query retrieved records for employees in either the Finance or Sales department. It demonstrated filtering based on multiple conditions using the `OR` operator.

Retrieve all employees not in IT: This task involved retrieving information about employees not belonging to the Information Technology (IT) department. The SQL query utilized the `NOT` operator to exclude records with the specified department.

Overall, these tasks illustrated the use of SQL queries to filter and retrieve specific information from database tables based on various conditions. Each task addressed different filtering requirements using operators such as `AND`, `OR`, and `NOT`, showcasing the versatility of SQL in data retrieval and analysis.