

Employee Database Management Using SQL Filters

Google Cybersecurity Professional Certification

| | |
|---|----------|
| Employee Database Management Using SQL Filters Project | 1 |
| Project description | 3 |
| Retrieve after hours failed login attempts | 4 |
| Retrieve login attempts on specific dates | 5 |
| Retrieve login attempts outside of Mexico | 6 |
| Retrieve employees in Marketing | 7 |
| Retrieve employees in Finance or Sales | 8 |
| Retrieve all employees not in IT | 9 |
| Summary | 10 |
| Importance of SQL in This Context: | 10 |

Project description

In this project, I utilized SQL to manage and filter through the employee database according to various scenarios. This project demonstrates the use of SQL to efficiently manage and query an employee database by applying conditional filters. Through various real-world scenarios, the project highlights the importance of these operators in retrieving specific subsets of data based on multiple criteria.

Retrieve after hours failed login attempts

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. Identify all failed login attempts that occurred after 18:00.

```
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
->
-> FROM log_in_attempts
->
-> WHERE login_time > '18:00' AND success = 0 ;
```

| 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 |
| 28 | aestrada | 2022-05-09 | 19:28:12 | MEXICO | 192.168.27.57 | 0 |
| 34 | drosas | 2022-05-11 | 21:02:04 | US | 192.168.45.93 | 0 |
| 42 | cgriffin | 2022-05-09 | 23:04:05 | US | 192.168.4.157 | 0 |
| 52 | cjackson | 2022-05-10 | 22:07:07 | CAN | 192.168.58.57 | 0 |
| 69 | wjaffrey | 2022-05-11 | 19:55:15 | USA | 192.168.100.17 | 0 |
| 82 | abernard | 2022-05-12 | 23:38:46 | MEX | 192.168.234.49 | 0 |
| 87 | apatel | 2022-05-08 | 22:38:31 | CANADA | 192.168.132.153 | 0 |
| 96 | ivelasco | 2022-05-09 | 22:36:36 | CAN | 192.168.84.194 | 0 |
| 104 | asundara | 2022-05-11 | 18:38:07 | US | 192.168.96.200 | 0 |
| 107 | bisles | 2022-05-12 | 20:25:57 | USA | 192.168.116.187 | 0 |
| 111 | aestrada | 2022-05-10 | 22:00:26 | MEXICO | 192.168.76.27 | 0 |
| 127 | abellmas | 2022-05-09 | 21:20:51 | CANADA | 192.168.70.122 | 0 |
| 131 | bisles | 2022-05-09 | 20:03:55 | US | 192.168.113.171 | 0 |
| 155 | cgriffin | 2022-05-12 | 22:18:42 | USA | 192.168.236.176 | 0 |
| 160 | jclark | 2022-05-10 | 20:49:00 | CANADA | 192.168.214.49 | 0 |
| 199 | yappiah | 2022-05-11 | 19:34:48 | MEXICO | 192.168.44.232 | 0 |

```
19 rows in set (0.235 sec)
```

To run this step, I used the following command:

```
Select *
From log_in_attempts
WHERE login_time > '18:00' AND success = 0 ;
```

This returned me with the results for 19 failed login attempts after hours.

Retrieve login attempts on specific dates

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.

```
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 | 1 |
| 3 | dkot | 2022-05-09 | 06:47:41 | USA | 192.168.151.162 | 1 |
| 4 | dkot | 2022-05-08 | 02:00:39 | USA | 192.168.178.71 | 0 |
| 8 | bisles | 2022-05-08 | 01:30:17 | US | 192.168.119.173 | 0 |
| 12 | dkot | 2022-05-08 | 09:11:34 | USA | 192.168.100.158 | 1 |
| 15 | lyamamot | 2022-05-09 | 17:17:26 | USA | 192.168.183.51 | 0 |
| 24 | arusso | 2022-05-09 | 06:49:39 | MEXICO | 192.168.171.192 | 1 |
| 25 | sbaelish | 2022-05-09 | 07:04:02 | US | 192.168.33.137 | 1 |
| 26 | apatel | 2022-05-08 | 17:27:00 | CANADA | 192.168.123.105 | 1 |
| 28 | aestrada | 2022-05-09 | 19:28:12 | MEXICO | 192.168.27.57 | 0 |
| 30 | yappiah | 2022-05-09 | 03:22:22 | MEX | 192.168.124.48 | 1 |
| 32 | acook | 2022-05-09 | 02:52:02 | CANADA | 192.168.142.239 | 0 |
| 36 | asundara | 2022-05-08 | 09:00:42 | US | 192.168.78.151 | 1 |
| 38 | sbaelish | 2022-05-09 | 14:40:01 | USA | 192.168.60.42 | 1 |
| 39 | yappiah | 2022-05-09 | 07:56:40 | MEXICO | 192.168.57.115 | 1 |
| 42 | cgriffin | 2022-05-09 | 23:04:05 | US | 192.168.4.157 | 0 |
| 43 | mcouliba | 2022-05-08 | 02:35:34 | CANADA | 192.168.16.208 | 0 |
| 44 | daquino | 2022-05-08 | 07:02:35 | CANADA | 192.168.168.144 | 0 |
| 47 | dkot | 2022-05-08 | 05:06:45 | US | 192.168.233.24 | 1 |
| 49 | asundara | 2022-05-08 | 14:00:01 | US | 192.168.173.213 | 0 |
| 53 | pmason | 2022-05-08 | 11:51:38 | CAN | 192.168.133.188 | 1 |

To run this step, I used the following command:

```
Select *  
From log_in_attempts  
WHERE login_date = '2022-05-09' OR login_date = '2022-05-08' ;
```

This returned me with the results for 75 login attempts made on these 2 dates.

Retrieve login attempts outside of Mexico

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.

```
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 | 1 |
| 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 | 1 |
| 4 | dkot | 2022-05-08 | 02:00:39 | USA | 192.168.178.71 | 0 |
| 5 | jrafael | 2022-05-11 | 03:05:59 | CANADA | 192.168.86.232 | 0 |
| 7 | eraab | 2022-05-11 | 01:45:14 | CAN | 192.168.170.243 | 1 |
| 8 | bisles | 2022-05-08 | 01:30:17 | US | 192.168.119.173 | 0 |
| 10 | jrafael | 2022-05-12 | 09:33:19 | CANADA | 192.168.228.221 | 0 |
| 11 | sgilmore | 2022-05-11 | 10:16:29 | CANADA | 192.168.140.81 | 0 |
| 12 | dkot | 2022-05-08 | 09:11:34 | USA | 192.168.100.158 | 1 |
| 13 | mrhah | 2022-05-11 | 09:29:34 | USA | 192.168.246.135 | 1 |
| 14 | sbaelish | 2022-05-10 | 10:20:18 | US | 192.168.16.99 | 1 |
| 15 | lyamamot | 2022-05-09 | 17:17:26 | USA | 192.168.183.51 | 0 |
| 16 | mcouliba | 2022-05-11 | 06:44:22 | CAN | 192.168.172.189 | 1 |
| 17 | pwashing | 2022-05-11 | 02:33:02 | USA | 192.168.81.89 | 1 |
| 18 | pwashing | 2022-05-11 | 19:28:50 | US | 192.168.66.142 | 0 |
| 19 | jhill | 2022-05-12 | 13:09:04 | US | 192.168.142.245 | 1 |
| 21 | iuduike | 2022-05-11 | 17:50:00 | US | 192.168.131.147 | 1 |
| 25 | sbaelish | 2022-05-09 | 07:04:02 | US | 192.168.33.137 | 1 |
| 26 | apatel | 2022-05-08 | 17:27:00 | CANADA | 192.168.123.105 | 1 |
| 29 | bisles | 2022-05-11 | 01:21:22 | US | 192.168.85.186 | 0 |

To run this step, I used the following command:

```
Select *  
From log_in_attempts  
WHERE NOT country LIKE 'MEX%';
```

As the country field included entries with 'MEX' and 'MEXICO', I used the 'MEX%' query to filter accordingly. This returned me with 144 attempts made outside of Mexico.

Retrieve employees in Marketing

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. Identify all employees in the Marketing department for all offices in the East building.

```
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 | k865l965m233 | 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 (0.002 sec)
```

To run this step, I used the following command:

```
Select *  
From employees  
WHERE department = 'Marketing'  
AND office LIKE 'East%' ;
```

This provided me with information for all employees from Marketing in the East building.

Retrieve employees in Finance or Sales

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.

```
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 |
| 1009 | NULL | lrodriqu | Sales | South-134 |
| 1010 | k242l212m542 | jlsansky | Finance | South-109 |
| 1011 | l748m120n401 | drosas | Sales | South-292 |
| 1015 | p611q262r945 | jsoto | Finance | North-271 |
| 1017 | r550s824t230 | jclark | Finance | North-188 |
| 1018 | s310t540u653 | abellmas | Finance | North-403 |
| 1022 | w237x430y567 | arusso | Finance | West-465 |
| 1024 | y976z753a267 | iuduike | Sales | South-215 |
| 1025 | z381a365b233 | jhill | Sales | North-115 |
| 1029 | d336e475f676 | ivelasco | Finance | East-156 |
| 1035 | j236k303l245 | bisles | Sales | South-171 |
| 1039 | n253o917p623 | cjackson | Sales | East-378 |
| 1041 | p929q222r778 | cgriffin | Sales | North-208 |
| 1044 | q428t157u158 | tharner | Finance | West-415 |

To run this step, I used the following command:

```
Select *  
From employees  
WHERE department = 'Finance' OR department = 'Sales' ;
```


Retrieve all employees not in IT

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.

```
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 |
| 1003 | d394e816f943 | sgilmore | Finance | South-153 |
| 1004 | e218f877g788 | eraab | Human Resources | South-127 |
| 1005 | f551g340h864 | gesparza | Human Resources | South-366 |
| 1007 | h174i497j413 | wjaffrey | Finance | North-406 |
| 1008 | i858j583k571 | abernard | Finance | South-170 |
| 1009 | NULL | lrodriqu | Sales | South-134 |
| 1010 | k242l212m542 | jlansky | Finance | South-109 |
| 1011 | l748m120n401 | drosas | Sales | South-292 |
| 1015 | p611q262r945 | jsoto | Finance | North-271 |
| 1016 | q793r736s288 | sbaelish | Human Resources | North-229 |
| 1017 | r550s824t230 | jclark | Finance | North-188 |
| 1018 | s310t540u653 | abellmas | Finance | North-403 |
| 1020 | u899v381w363 | arutley | Marketing | South-351 |
| 1022 | w237x430y567 | arusso | Finance | West-465 |
| 1024 | y976z753a267 | iuduike | Sales | South-215 |
| 1025 | z381a365b233 | jhill | Sales | North-115 |
| 1026 | a998b568c863 | apatel | Human Resources | West-320 |

Similarly, I used the following command:

```
Select *  
From employees  
WHERE NOT department = 'Information Technology' ;
```

Summary

This project showcases the power of SQL in filtering and analyzing data to meet specific business requirements. By applying filters such as date ranges, geographical restrictions, and conditional logic, we effectively retrieve meaningful insights from complex datasets. The scenarios explored include identifying failed login attempts during after-hours, isolating login activities based on dates or locations, and segmenting employees by departments. These use cases highlight the importance of SQL in ensuring data-driven decisions, enhancing security, and streamlining organizational processes.

Importance of SQL in This Context:

- **Efficiency:** SQL enables quick and accurate data retrieval from large datasets, essential for informed decision-making in organizations.
- **Flexibility:** The use of **AND**, **OR**, and **NOT** allows complex queries to be crafted with precision, making it possible to address nuanced business requirements.
- **Scalability:** These SQL filters work seamlessly across large-scale databases, ensuring scalability as the organization grows.