

Apply filters to SQL queries

Project description

My organization is working to make their system more secure. It is my job to ensure the system is safe, investigate all potential security issues, and update employee computers as needed. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

Retrieve after hours failed login attempts

All login attempts that failed after business hours (18:00) need to be investigated.

The following code demonstrates how I created a SQL query to filter for failed login attempts that occurred after business hours.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
-> FROM log_in_attempts
-> WHERE login_time > '18:00' AND success = FALSE;
```

This query filters for failed login attempts that occurred after 18:00. First, I started by selecting all data from the `log_in_attempts` table. Then, I used a `WHERE` clause with an `AND` operator to filter my results to output only login attempts that occurred after 18:00 and were unsuccessful. The first condition is `login_time > '18:00'`, which filters for the login attempts that occurred after 18:00. The second condition is `success = FALSE`, which filters for the failed login attempts.

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 (0.001 sec)

The output is that there were a total of 19 failed attempts after work hours (18:00).

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. All login events that happened on 2022-05-09 or on the day before needs to be investigated.

The following code demonstrates how I created a SQL query to filter for login attempts that occurred on specific dates.

```
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

This query filters for failed login attempts that occurred on 2022-05-09 or 2022-05-08. First, I started by selecting all data from the log_in_attempts table. Then, I used a WHERE clause with an OR operator to filter my results to output only login attempts that occurred on either 2022-05-09 or 2022-05-08.

```
162 | yappiah | 2022-05-09 | 04:51:22 | MEXICO | 192.168.162.100 | 0 |
163 | tmitchel | 2022-05-08 | 09:21:16 | MEX | 192.168.119.29 | 0 |
165 | jreckley | 2022-05-08 | 15:28:43 | MEXICO | 192.168.34.193 | 0 |
168 | jlansky | 2022-05-08 | 13:25:42 | USA | 192.168.210.94 | 1 |
169 | alevitsk | 2022-05-08 | 08:10:43 | CANADA | 192.168.210.228 | 0 |
170 | sbaelish | 2022-05-09 | 16:43:18 | USA | 192.168.65.113 | 0 |
172 | mabadi | 2022-05-08 | 08:06:50 | US | 192.168.180.41 | 1 |
178 | sgilmore | 2022-05-08 | 12:27:22 | CAN | 192.168.52.216 | 0 |
184 | alevitsk | 2022-05-08 | 03:09:48 | CAN | 192.168.33.70 | 0 |
186 | bisles | 2022-05-09 | 04:29:17 | USA | 192.168.40.72 | 0 |
187 | arusso | 2022-05-09 | 00:36:26 | MEX | 192.168.77.137 | 0 |
189 | nmason | 2022-05-08 | 05:37:24 | CANADA | 192.168.168.117 | 1 |
190 | jsoto | 2022-05-09 | 05:09:21 | USA | 192.168.25.60 | 0 |
191 | cjackson | 2022-05-08 | 06:46:07 | CANADA | 192.168.7.187 | 0 |
193 | lrodrigu | 2022-05-08 | 07:11:29 | US | 192.168.125.240 | 0 |
197 | jsoto | 2022-05-08 | 09:05:09 | US | 192.168.36.21 | 0 |
```

75 rows in set (0.001 sec)

```
MariaDB [organization]>
```

The output is that there were a total of 75 login attempts on 2022-05-09 and 2022-05-08.

Retrieve login attempts outside of Mexico

There are issues with the login attempts that occurred outside of Mexico that must be investigated.

The following code demonstrates how I created a SQL query to filter for 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

This query filters all login attempts that occurred in countries other than Mexico. First, I started by selecting all data from the `log_in_attempts` table. Then, I used a `WHERE` clause with `NOT` to filter for countries other than Mexico. I used `LIKE` with `MEX%` as the pattern to match because the dataset represents Mexico as `MEX` and `MEXICO`.

```
178 | sgilmore | 2022-05-08 | 12:27:22 | CAN | 192.168.92.210 | 0  
179 | jclark | 2022-05-12 | 04:08:17 | CAN | 192.168.232.93 | 0  
181 | abellmas | 2022-05-10 | 13:37:05 | CAN | 192.168.60.111 | 0  
182 | lyamamot | 2022-05-10 | 06:01:31 | USA | 192.168.106.52 | 0  
183 | nmason | 2022-05-11 | 05:29:36 | CANADA | 192.168.137.147 | 0  
184 | alevitsk | 2022-05-08 | 03:09:48 | CAN | 192.168.33.70 | 0  
185 | jsoto | 2022-05-10 | 13:34:58 | USA | 192.168.151.91 | 0  
186 | bisles | 2022-05-09 | 04:29:17 | USA | 192.168.40.72 | 0  
188 | jsoto | 2022-05-11 | 00:39:09 | USA | 192.168.21.88 | 0  
189 | nmason | 2022-05-08 | 05:37:24 | CANADA | 192.168.168.117 | 1  
190 | jsoto | 2022-05-09 | 05:09:21 | USA | 192.168.25.60 | 0  
191 | cjackson | 2022-05-08 | 06:46:07 | CANADA | 192.168.7.187 | 0  
192 | bisles | 2022-05-10 | 08:32:03 | USA | 192.168.201.40 | 1  
193 | lrodriqu | 2022-05-08 | 07:11:29 | US | 192.168.125.240 | 0  
194 | jclark | 2022-05-12 | 14:11:04 | CAN | 192.168.197.247 | 0  
195 | alevitsk | 2022-05-11 | 06:59:13 | CANADA | 192.168.236.78 | 1  
196 | acook | 2022-05-10 | 09:56:48 | CAN | 192.168.52.90 | 0  
197 | jsoto | 2022-05-08 | 09:05:09 | US | 192.168.36.21 | 0  
200 | jclark | 2022-05-12 | 01:11:45 | CANADA | 192.168.91.103 | 1
```

```
144 rows in set (0.001 sec)  
  
MariaDB [organization]> 
```

The output is 144 login attempts in other countries other than Mexico.

Retrieve employees in Marketing

My team wants to update the computers for the Marketing employees.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Marketing department.

```
MariaDB [organization]> SELECT *  
  -> FROM employees  
  -> WHERE department = 'Marketing';
```

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1001	b239c825d303	bmoreno	Marketing	Central-276

This query returns all employees in the Marketing department. First, I started by selecting all data from the **employees** table. Then, I used a **WHERE** clause to filter for employees who work in the Marketing department.

1167	1738m922n515	tblackwe	Marketing	North-443
1172	q372r826s628	akhan	Marketing	Central-360
1173	r537s849t690	ialcazar	Marketing	South-429
1178	w986x187y885	nlannist	Marketing	North-196
1190	NULL	kcarter	Marketing	Central-270
1191	NULL	shakimi	Marketing	Central-366
1198	q308r573s459	jmartine	Marketing	South-117

```
44 rows in set (0.001 sec)  
  
MariaDB [organization]> █
```

The output is that there are 44 employees in the Marketing department that need their devices updated.

Retrieve employees in Finance or Sales

The machines for employees in the Finance and Sales departments also need to be updated. Since a different security update is needed, I have to get information on employees only from these two departments.

The following code demonstrates how I created a SQL query to filter for employee machines from employees in the Finance or Sales 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	jlansky	Finance	South-109

This query returns all employees in the Finance and Sales departments. First, I started by selecting all data from the **employees** table. Then, I used a **WHERE** clause with **OR** to filter for employees who are in the Finance and Sales departments. The first condition is **department = 'Finance'**, which filters for employees from the Finance department. The second condition is **department = 'Sales'**, which filters for employees from the Sales department.

1181	z803a233b718	sessa	Finance	South-207
1185	d790e839f461	revens	Sales	North-330
1186	e281f433g404	sacosta	Sales	North-460
1187	f963g637h851	bbode	Finance	East-351
1188	g164h566i795	noshiro	Finance	West-252
1195	n516o853p957	orainier	Finance	East-346

```
71 rows in set (0.001 sec)  
  
MariaDB [organization]>
```

The output is that there are 71 employees from the Finance and Sales departments that need their devices updated.

Retrieve all employees not in IT

My team needs to make one more security update on employees who are not in the Information Technology department.

The following demonstrates how I created a SQL query to filter for employee machines from employees not in the Information Technology department.

```
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

The query returns all employees not in the Information Technology department. First, I started by selecting all data from the **employees** table. Then, I used a **WHERE** clause with **NOT** to filter for employees not in this department.

1194	m340n287o441	zwarren	Human Resources	West-212
1195	n516o853p957	orainier	Finance	East-346
1198	q308r573s459	jmartine	Marketing	South-117
1199	r520s571t459	areyes	Human Resources	East-100

```
161 rows in set (0.001 sec)  
  
MariaDB [organization]>
```

The output is that there are 161 employees from all the departments other than IT that will get additional updates on their devices.

Summary

I applied filters to SQL queries to get specific information on login attempts and employees from different departments. I used two different tables, **log_in_attempts** and **employees**. In addition, I used the **AND**, **OR**, and **NOT** operators to filter for the specific information needed for each task and **LIKE** and the percentage sign **(%)** wildcard to filter for patterns.