

Apply filters to SQL queries

Project description

Being a security professional, I was given a task by my organisation to investigate security issues to help secure the system. I discovered issues with login attempts and employees' machines. So I referred to the **employees** and **log_in_attempts** tables on my organisation's database to retrieve datasets that could help in identifying this risk. The following SQL filters were used to collect dataset and related records for my investigations:

Retrieve after hours failed login attempts

```
clear
MariaDB [organization]> SELECT *
->   FROM log_in_attempts
-> WHERE login_time > '18:00' AND success = 0
-> ORDER BY login_time;
+-----+-----+-----+-----+-----+-----+
| event_id | username | login_date | login_time | country | ip_address | success |
+-----+-----+-----+-----+-----+-----+
| 104 | asundara | 2022-05-11 | 18:38:07 | US | 192.168.96.200 | 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 |
| 18 | pwashing | 2022-05-11 | 19:28:50 | US | 192.168.66.142 | 0 |
| 199 | yappiah | 2022-05-11 | 19:34:48 | MEXICO | 192.168.44.232 | 0 |
| 69 | wjaffrey | 2022-05-11 | 19:55:15 | USA | 192.168.100.17 | 0 |
| 131 | bisles | 2022-05-09 | 20:03:55 | US | 192.168.113.171 | 0 |
| 107 | bisles | 2022-05-12 | 20:25:57 | USA | 192.168.116.187 | 0 |
| 2 | apatel | 2022-05-10 | 20:27:27 | CAN | 192.168.205.12 | 0 |
| 160 | jclark | 2022-05-10 | 20:49:00 | CANADA | 192.168.214.49 | 0 |
| 34 | drosas | 2022-05-11 | 21:02:04 | US | 192.168.45.93 | 0 |
| 127 | abellmas | 2022-05-09 | 21:20:51 | CANADA | 192.168.70.122 | 0 |
| 111 | aestrada | 2022-05-10 | 22:00:26 | MEXICO | 192.168.76.27 | 0 |
| 52 | cjackson | 2022-05-10 | 22:07:07 | CAN | 192.168.58.57 | 0 |
| 155 | cgriffin | 2022-05-12 | 22:18:42 | USA | 192.168.236.176 | 0 |
| 96 | ivelasco | 2022-05-09 | 22:36:36 | CAN | 192.168.84.194 | 0 |
| 87 | apatel | 2022-05-08 | 22:38:31 | CANADA | 192.168.132.153 | 0 |
| 42 | cgriffin | 2022-05-09 | 23:04:05 | US | 192.168.4.157 | 0 |
| 82 | abernard | 2022-05-12 | 23:38:46 | MEX | 192.168.234.49 | 0 |
+-----+-----+-----+-----+-----+-----+
19 rows in set (0.023 sec)

MariaDB [organization]>
```

The organisation's working hours end by 18:00 and as such, working activities close by that time which means that every employee should have logged out from the system on or before the specified time. But it was observed that some unusual activities went on after working hours. So, further investigation had to be carried out. So I queried the database to filter login attempts that were made after the closing hours. So I queried the database with the instruction; **SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND success = 0 ORDER BY login_time;**

The instruction above queries the database to return all information from the `log_in_attempt` table with only the login attempts made after **18:00** and where the success is recorded as **0**. The **AND** operator was used also to specify that both conditions had to be met.

NOTE: `login_time` is recorded as string, while the success is recorded as boolean which only returns true or false (i.e 1 means true and 0 means false). The information is returned as shown in the screenshot above.

Retrieve login attempts on specific dates

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	daguino	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	nmason	2022-05-08	11:51:38	CAN	192.168.133.188	1
56	acook	2022-05-08	04:56:30	CAN	192.168.209.130	1
58	ivelasco	2022-05-09	17:20:54	CAN	192.168.57.162	0
61	dтанака	2022-05-09	09:45:18	USA	192.168.98.221	1
65	aalonso	2022-05-09	23:42:12	MEX	192.168.52.37	1
66	aestrada	2022-05-08	21:58:32	MEX	192.168.67.223	1
67	abernard	2022-05-09	11:53:41	MEX	192.168.118.29	1
68	mrah	2022-05-08	17:16:13	US	192.168.42.248	1
70	tmitchel	2022-05-09	10:55:17	MEXICO	192.168.87.199	1

A suspicious event occurred in the organisation on 2022-05-09. So to investigate the event, I queried the database to display all login attempts for 2022-05-09 or 2022-05-08. I used the following query to filter all login attempts from the `log_in_attempts` table;

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

The query above filters the database to return all `login_attempts` made on the 2022-05-09 OR 2022-05-08. The **OR** operator specifies that any of the conditions can be met provided that either of them are true. The information is returned as shown in the screenshot above.

Retrieve login attempts outside of Mexico

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	mrah	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
31	acook	2022-05-12	17:36:45	CANADA	192.168.58.232	0
32	acook	2022-05-09	02:52:02	CANADA	192.168.142.239	0
33	zbernal	2022-05-11	02:52:10	US	192.168.72.59	1
34	drosas	2022-05-11	21:02:04	US	192.168.45.93	0
36	asundara	2022-05-08	09:00:42	US	192.168.78.151	1
37	eraab	2022-05-10	06:03:41	CANADA	192.168.152.148	0
38	sbaelish	2022-05-09	14:40:01	USA	192.168.60.42	1
41	apatel	2022-05-10	17:39:42	CANADA	192.168.46.207	0

There have been suspicious activities with login attempts, but it was noticed that none of these suspicions are coming from mexico. So I filtered the query to ensure that Mexico has been excluded from the investigation. The query is shown as follows; **SELECT * FROM log_in_attempts WHERE NOT country LIKE 'MEX%'**;

The **NOT** was used to remove Mexico from the search and to make sure that no form of Mexico like “mex” was returned by the database, I used the **LIKE** keyword to ensure that the database would not return any country that is abbreviated as “mex”.the result returned is shown on the screenshot above.

Retrieve employees in Marketing

```
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 | k8651965m233 | 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 (0.001 sec)
```

The team was looking to perform some security updates on some employees' machines in the department of marketing whose offices are located in the east building. I was saddled with the responsibility of querying the **employees** table in the database in order to get the specified information. I had to query the database to filter out the information that was needed by my team for the security updates. I used the following query; **SELECT * FROM employees WHERE department = 'marketing' AND office LIKE 'east%'**;

The above query filters every information that has to do with the marketing department from the employee's table in the database. The **AND** operator was used to check for the conditions to ensure that both were true. The **LIKE** keyword was also used to return all office numbers in the east building. The information returned is shown in the screenshot above.

Retrieve employees in Finance or 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 |
| 1009 | NULL | lrodriqu | Sales | South-134 |
| 1010 | k2421212m542 | jlansky | Finance | South-109 |
| 1011 | 1748m120n401 | 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 | s429t157u159 | tbarnes | Finance | West-415 |
| 1045 | t567u844v434 | pwashing | Finance | East-115 |
| 1046 | u429v921w138 | daquino | Finance | West-280 |
| 1047 | v109w587x644 | cward | Finance | West-373 |
| 1048 | w167x592y375 | tmitchel | Finance | South-288 |
| 1049 | NULL | jreckley | Finance | Central-295 |
| 1050 | y132z930a114 | csimmons | Finance | North-468 |
| 1057 | f370g535h632 | mscott | Sales | South-270 |
| 1062 | k3671639m697 | redwards | Finance | North-180 |
| 1063 | 1686m140n569 | lpope | Sales | East-226 |
| 1066 | o678p794q957 | ttyrell | Sales | Central-444 |
| 1069 | NULL | jpark | Finance | East-110 |
| 1071 | t244u829v723 | zdutchma | Sales | West-348 |
+-----+-----+-----+-----+-----+
```

The team had to perform another update on the employees machines in the finance and sales department. So I used a query to filter the database to return only the information for the sales and finance department from the employees table. So i used the query;

SELECT * FROM employees WHERE department = 'finance' OR department = 'sales';

The query above filters all information relating to the finance or the sales department from the employees table in the database. The **OR** operator specifies that either condition can be met.which means that the department information returned should either be for sales or finance. The information returned is shown in the screenshot above.

Retrieve all employees not in IT

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	sgilmores	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	lrodriguez	Sales	South-134
1010	k2421212m542	jlancky	Finance	South-109
1011	l748m120n401	drosas	Sales	South-292
1015	p611q262r945	jsoto	Finance	North-271
1016	q793z736s288	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	iuduiken	Sales	South-215
1025	z381a365b233	jhill	Sales	North-115
1026	a998b568c863	apatel	Human Resources	West-320
1027	b806c503d354	mrah	Marketing	West-246
1028	c603d749e374	aestrada	Human Resources	West-121
1029	d336e475f676	ivelasco	Finance	East-156
1030	e391f189g913	mabadi	Marketing	West-375
1031	f419g188h578	dkot	Marketing	West-408
1034	i679j565k940	bsand	Human Resources	East-484
1035	j236k303l245	bisles	Sales	South-171
1036	k550l533m205	rjensen	Marketing	Central-239
1038	m873n636o225	btang	Human Resources	Central-260
1039	n253o917p623	cjackson	Sales	East-378
1040	o783p832q294	dtarly	Human Resources	East-237
1041	p929q222r778	cgriffin	Sales	North-208
1042	q175r338s833	acook	Human Resources	West-381

The team has to make updates on employees machine for all departments. But it was noticed that the information technology(IT)department has this update already. So I had to query the database to return information from all departments excluding the IT department. I used the following filter to query the department;

SELECT * FROM employees WHERE NOT department = 'information technology';

The query above returns all information for all departments excluding the information technology department. The **NOT** operator negates a condition. Which means that it returns all other departments information except the IT department. The information returned is shown in the screenshot above.

Summary

During the course of every investigation carried out, I used SQL filters to query the database to return specific information from specific tables. I used logical operators such as **AND**, **OR** and **NOT** operators to filter out specific information from the tables of the database and to make sure that only the needed information is returned.