

# Apply Filters to SQL Queries

## Apply Filters to SQL Queries

### Project description

As a security professional at a large organization, I am responsible for investigating potential security issues to ensure the system remains secure. I have been informed of concerns related to login attempts and employee machines. My task is to examine the organization's data by querying the `employees` and `log_in_attempts` tables using SQL filters. By retrieving relevant records from these datasets, I used the query outputs and analyzed login patterns to identify any suspicious activity to mitigate potential threats.

### Retrieve after hours failed login attempts

```
MariaDB [organization]> clear
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
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 (0.147 sec)

Company business hours end at 6 pm (18:00) and I was tasked to look into failed login attempts made after business hours in a potential security incident. To investigate this, I made a sequel query where I asked the database to give me all the information (`SELECT *`) of accounts that tried to log in (`FROM log_in_attempts`) after business hours (`WHERE login_time > '18:00'`) and who failed (`AND success = FALSE`). The query gave me back an output of information about

# Apply Filters to SQL Queries

failed login attempts, giving me the `event_id`, `username` of the account that failed to log in, the `login_date` and `login_time` of the attempt, the `country` in which the attempt was made, the `ip_address` of the device used to attempt, and `success` of the attempt. The success of the attempts are shown as 0 to indicate a FALSE success, if the individual had succeeded then the success column would have a 1 to indicate a TRUE success, but since I only specified login attempts that failed there are no 1s in the results. The AND in (WHERE `login_time` > '18:00' AND `success` = FALSE lets SQL know that both conditions needed to be met in the results I needed which is how I got the specific results.

## Retrieve login attempts 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
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	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	dtanaka	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	mrh	2022-05-08	17:16:13	US	192.168.42.248	1

A suspicious event occurred on 2022-05-09. To investigate this event, I was asked to look into the attempts made in the days of 05/08/2022 and 05/09/2022 so I then updated the query to show me specifically `log_in_attempts` made within the `login_date` - '2022-05-09' OR `login_date` - '2022-05-08' since using AND would only give the information of the accounts

# Apply Filters to SQL Queries

that made the attempts on both days and not just ones that attempted on one and not the other or even on both days. By specifying the dates I need I can avoid having to go scrolling through hundreds of results trying to find the dates I need.

## Retrieve login attempts 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
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's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. I had to make a SQL query to investigate login attempts that occurred outside of Mexico. I kept the same first two command lines since I still am requesting all the information of the account and events of the table `log_in_attempts`, only this time I changed the `WHERE` to indicate that I was looking for results that were `NOT` from the `country` of Mexico, but different tables files showed Mexico written as `MEXICO` and some as `MEX`, so I had to make the command line read `WHERE NOT country LIKE 'MEX%'`. The % sign after Mex indicates I'm looking for results where the `country` column has results that have `MEX` in it but that also could have more letters after the `MEX`; like `ICO` for example. This would result in the query giving

# Apply Filters to SQL Queries

me results from all attempts made outside of Mexico, so the query would isolate any results that have **MEX** or **MEXICO** or any variation of Mexico in the **country** column.

## 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 | 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 (0.001 sec)

MariaDB [organization]> 
```

My team wants to perform security updates on specific employee machines in the **Marketing** department who work in all **East** offices. I am responsible for getting information on these employee machines, so I made a query on information **FROM** the **employees** table this time looking for people who work in the **Marketing** department **AND** who work in any of the **East** offices. Before, I was inquiring for information on the **log\_in\_attempts** table but now I'm looking for information on **employees** so the **FROM** now reads **FROM employees** to reflect the change in table. I'm still looking to get all the information on the results so the **SELECT \*** stays the same. Since I'm looking for results for employees in the **Marketing** department, and who work in the **East** offices the **WHERE** then had to be adjusted to reflect my query and so I wrote the query as **WHERE department = 'Marketing' AND office LIKE 'East%'**. This line indicates that I'm seeking results that have the **Marketing** value in the **department** column but also that the results must contain a value of **East** in the **office** column but the **%** sign indicates that it could have anything after **East** allowing the results to come back with every **employee** from every possible **East** office available.

# Apply Filters to SQL Queries

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	k242l212m542	jlansky	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	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	k367l639m697	redwards	Finance	North-180
1063	l686m140n569	lpope	Sales	East-226
1066	o678p794q957	ttyrell	Sales	Central-444
1069	NULL	jpark	Finance	East-110
1071	t244u829v723	zdutchma	Sales	West-348

My team now needs to perform a different security update on machines for employees in the Sales and Finance departments. To get the information I would need I went ahead and made a query that gave me results for the machines that are in the Finance department and the Sales department by just altering the WHERE to specifically ask for the two departments I need. The first two lines would remain the same from the previous query I made but the WHERE would now read WHERE department = 'Finance' OR department = 'Sales' giving me a list of all employees that are either in Finance or in Sales.



# Apply Filters to SQL Queries

Retrieve all employees not in 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
1027	b806c503d354	mrah	Marketing	West-246
1028	c603d749e374	astrada	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

My 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. The query would then have to isolate machines that are in the **Information Technology** department and to do this I wrote the query to indicate im not looking for machines that have the value of **Information Technology** in the **department** column by having it read **WHERE NOT department = 'Information Technology'**. This gave me all the results for every department aside from IT.

## Summary

I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, **log\_in\_attempts** and **employees**. I used the **AND**, **OR**, and **NOT** operators to filter for the specific information needed for each task. I also used **LIKE** and the percentage sign (%) wildcard to filter for patterns.