

Apply filters to SQL queries

Project description

The purpose of this project is to find information in tables about employees, and their machines and departments to investigate suspicious login activity as well as update computers. Using keywords such as NOT, AND, OR, and operators will assist in filtering for the information needed.

Retrieve after hours failed login attempts

In the scenario below, I was supposed to investigate failed login attempts after the time of 6 PM. I used the AND keyword to accomplish this task.

```
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	astrada	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	astrada	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 (0.153 sec)

MariaDB [organization]>
```

Retrieve login attempts on specific dates

In the scenario below I was tasked to investigate a suspicious event that happened within the company. I had to look at login attempts on 2022-05-09 as well as the day before. For this, I used the `OR` keyword to filter the data.

```
MariaDB [organization]> SELECT *  
  -> FROM log_in_attempts  
  -> WHERE login_date = '2022-05-08' OR login_date = '2022-05-09';
```

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

Retrieve login attempts outside of Mexico

In this example, I was instructed to investigate the login attempts that were not in Mexico. I used the `NOT` keyword to get login information outside of Mexico. I also used `LIKE` and `%` operator to take into account that the abbreviation “MEX” could be within the column values. `%` is a wildcard operator that can account for more than one character after “MEX”, so that the output can be either “MEX” or “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	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

Retrieve employees in Marketing

In the scenario below, my team and I had to update machines for all Marketing employees within the East-located offices of the company. I used the `AND` keyword and `LIKE` and `%` operator to receive the output below.

```
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 (0.040 sec)

Retrieve employees in Finance or Sales

In this example, my team had to update computers to all employees in either the Finance or the Sales department. To gather information about these employees, I used the **OR** keyword to filter and gather data.

```
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	iuduiky	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

Retrieve all employees not in IT

In this scenario the company instructed me to update computers for all the employees that are not in the IT department. To find info about all non-IT employees, I used the **NOT** keyword in the form of **!=** in this query.

```
MariaDB [organization]> SELECT *
```

```
-> FROM employees
```

```
-> WHERE 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

Summary

I gathered specific information using keywords such as **AND**, **OR**, and **NOT** and operators such as **<**, **>**, etc. to narrow down the output. This information was used to investigate suspicious login attempts at different times and locations. The information was also used to see which employees need their computers updated.