

Apply filters to SQL queries

Project description

Our organization's team requires data to investigate potential security issues and update computers. The objective of this project is to utilize SQL filtering to extract the necessary information from the database. Initially, we will retrieve all failed login attempts outside of business hours. Subsequently, we will obtain all login attempts made on specific dates. Additionally, we will identify logins that did not originate in Mexico. Furthermore, we will collect data about employees within the Marketing department. Moreover, we will gather information pertaining to employees in the Finance or Sales departments. Lastly, we will acquire details regarding employees who are not part of the Information Technology department.

Retrieve after hours failed login attempts

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

First we want to use a query that selects information from the log_in_attempts table and apply a filter that will show us the log-in-attempts made after hours in the organization. “After-hours” is considered any time after 6pm, so we will input that we only want to see information from the login_time column after ‘18:00:00’. Since we also only want those results to be from failed attempts, we will apply the filter that only shows us where success is false, and since the values of True and False are Boolean values in this case, we want to filter the success column for false (which is the number ‘0’.) This returns the following:

```
+-----+-----+-----+-----+-----+-----+-----+
| 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 (0.036 sec)

MariaDB [organization]> 
```

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';
```

Our team is investigating a suspicious event that occurred on '2022-05-09'. We want to retrieve all login attempts that occurred on this day and the day prior, which is '2022-05-08'. To do this, we want to select the login_date column information from the log_in_attempts table containing information on the dates when the login attempts were made. Using the query above, the following is returned:

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	astrada	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	astrada	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
71	mcouliba	2022-05-09	06:57:42	CAN	192.168.55.169	0
72	alevitsk	2022-05-08	12:09:10	CANADA	192.168.139.176	1
79	abernard	2022-05-09	11:41:15	MEX	192.168.158.170	0
80	cjackson	2022-05-08	02:18:10	CANADA	192.168.33.140	1
83	lrodriqu	2022-05-08	08:10:23	USA	192.168.67.69	1
87	apatel	2022-05-08	22:38:31	CANADA	192.168.132.153	0
90	gesparza	2022-05-09	00:49:05	CANADA	192.168.87.201	0
92	pwashing	2022-05-08	00:36:12	US	192.168.247.219	0
96	ivelasco	2022-05-09	22:36:36	CAN	192.168.84.194	0
97	jreckley	2022-05-09	02:49:23	MEXICO	192.168.32.231	1
101	sbaelish	2022-05-08	12:01:22	US	192.168.145.158	0
102	jreckley	2022-05-09	16:51:44	MEX	192.168.108.13	1
108	daquino	2022-05-09	21:30:48	CANADA	192.168.15.110	1
110	mabadi	2022-05-09	00:01:54	USA	192.168.90.124	1
112	rjensen	2022-05-09	09:22:05	MEX	192.168.69.116	1
117	bsand	2022-05-08	00:19:11	USA	192.168.197.187	0
120	tmitchel	2022-05-09	02:58:17	MEXICO	192.168.134.62	0

Retrieve login attempts outside of Mexico

```
MariaDB [organization]> SELECT *  
-> FROM log_in_attempts  
-> WHERE NOT country LIKE 'MEX%';
```

Now, our team is investigating logins that did not originate in Mexico, and we need to find this information. We will be utilizing the NOT and LIKE operators to complete this query. The NOT operator will show us any information that is not the same as the indicator we type in. The LIKE operator uses wildcards to filter for specific characters or numbers in a string.

The query above returns the following:

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	mrh	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	iuduke	2022-05-11	17:50:00	US	192.168.131.147	1

Retrieve employees in Marketing

To retrieve the employees in marketing, we need to query the database for information from the employees table. More specifically, the department and office columns, but we will start with the table itself and find it returned the table to us like below:

```
MariaDB [organization]> SELECT *
-> FROM employees;
```

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1001	b239e825d303	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
1006	g329h357i597	alevitsk	Information Technology	East-320
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
1012	m756n668o146	nmason	Information Technology	North-160
1013	n205o559p243	zbernal	Information Technology	South-229
1014	NULL	asundara	Information Technology	West-219
1015	p611q262r945	jsoto	Finance	North-271
1016	q793r736s288	sbaelish	Human Resources	North-229
1017	r550s824t230	jclark	Finance	North-188

Since we are updating employee machines, and need to obtain the information about employees in the 'Marketing' department who are located in all offices of the East Building (such as 'East-171' or 'East-322'), we will use the AND and LIKE operators to filter for those both of these conditions.

```
MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE department = 'Marketing' AND office LIKE 'East-%';
```


Using the AND operator, we are able to filter for more than one condition simultaneously, and using the LIKE operator, we are able to filter for all offices in the East building by omitting the numbers of the offices. The following is returned by entering the query:

```

+-----+-----+-----+-----+-----+
| 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 | randers | Marketing | East-460 |
| 1156 | a184b775c707 | dellery | Marketing | East-417 |
| 1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.005 sec)

MariaDB [organization]> 

```

Retrieve employees in Finance or Sales

Now, our team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and we need to locate information on these employees. To do so we will write a query to retrieve records for employees in the 'Finance' or the 'Sales' department.

```

MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE department = 'Sales' OR department = 'Finance';

```

The OR operator allows us to meet 2 separate conditions in the same column as you can see in the return below:

```

+-----+-----+-----+-----+-----+
| 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 | x550s824t230 | 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 | x567v844w454 | wubling | Finance | East-115 |
+-----+-----+-----+-----+-----+

```

Retrieve all employees not in IT

Our team needs to make one more update. This update has already been made to employee computers in the Information Technology department. The team needs information about employees who are not in that department. To do this, we will use the NOT operator to identify these employees. The NOT operator is placed in front of the condition to indicate that we are NOT selecting the information that falls under the condition's criteria like below.

```

MariaDB [organization]> clear
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 |

```

Summary

Summarily, we have used the AND operator to find unsuccessful login attempts made after organizational hours by reading Boolean data. Then, we used the OR operator to retrieve login attempts from specific dates. We used the LIKE operator to apply a filter that retrieved login attempts outside of Mexico and coupled the LIKE operator with the AND operator to satisfy 2 criteria at once, which found us information about employees who worked not only in a certain building, but also in a specific department. Also, we used the OR operator to retrieve employees in 2 departments at once. Finally, we used the NOT operator to retrieve employees in all departments except for IT.