

# Apply Filters to SQL queries

## Project Description

In this project, I need to use SQL commands to analyze data and retrieve specific information from a database.

When running SQL queries, using the correct operators to filter the information you're looking for in a table is essential to navigate databases. I will type in commands with the **AND**, **OR**, and **NOT** operators to look for the necessary information.

Here is the scenario:

I will obtain specific information about employees, their machines, and the departments they belong to from this **MariaDB** database using SQL commands.

## Retrieve After-Hours Failed Login Attempts

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

To start working on this database, I will query failed login attempts, specifically all unsuccessful attempts after 18:00. For this command, I will need to use the **SELECT** keyword with an asterisk (\*) to select all the columns in the table. From there, we will use the **FROM** keyword to indicate which table we will query, hence the name **log\_in\_attempts**.

After that, we will then filter the information by starting with the **WHERE** keyword to indicate the condition for a filter and use the columns (**login\_time** and **success**) to specify our filter. To look for the login attempts that have failed after **18:00**, we will use the greater than sign (>) to indicate that the value in the **login\_time** column must be greater than. Since 18:00 is a string data type, we will put it in quotation marks.

Lastly, we will use the **AND** operator to state that not only do we want the login attempts after 18:00 but we also want all of the failed attempts together. **FALSE** is a boolean data type, which is stored as either a 0 (false) or a 1 (true) - in this case, we want to look for all of the false conditions in the success column.

Here is our output from the command:

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 (0.250 sec)

## 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';
```

For this instance, we want to retrieve all login attempts that occurred on **'2022-05-09'** and the day before (**'2022-05-08'**). Since we know about the **SELECT** and **FROM** keywords now, let's go straight to the filter portion of this command.

After the **WHERE** keyword, we will use the **login\_date** column to specify the login dates we want to compare. We will put our first date in quotes, to signify that it is a string data type, and then place our **OR** operator in between our log-in date comparison. The **OR** operator specifies that either condition can be met or both of them can be met. After the operator, we'll put in our other login date by specifying the column.

Here is the output of this command:

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	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

## Retrieve Login Attempts Outside of Mexico

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

In the query, we want to look at unsuccessful login attempts in Mexico. Since there are field entries such as 'MEX' and 'MEXICO' we need to use the percentage sign (%). The percentage sign will be recognized as a wildcard, this will substitute any number of other characters after it. To find matching patterns, we should use 'MEX%' to find all related entries.

For this filter, we'll use the **WHERE** keyword followed by the **NOT** keyword. The **NOT** keyword negates the condition of the filter, so this will fulfill the 'not in Mexico' characteristic of what

we're searching for. After that, we have our column **country**, where we will search through as well as the **LIKE** keyword following it.

The **LIKE** keyword is needed when dealing with wildcards, instead of the equal sign (=). This is because the equal sign is finding that exact match, whereas **LIKE** allows for pattern matching. Once we put '**MEX%**' in quotations, we can now search for all the countries that are not Mexico.

Here is the output of this command:

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
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

## Retrieve Employees in Marketing

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

We will use the **employees** table for the next few SQL queries. For this particular query, we will be looking at all of the employees in the '**Marketing**' department who are located in all the offices in the East building (an example of this would be '**East-216**')

If we take a look at our filter for this query, we'll start with our **WHERE** keyword and then choose the column **department** that equals '**Marketing**' for all the employees in this department. After that, we'll use our **AND** operator to include the '**office**' column to look for offices that match the East building.

Lastly, we'll add in our **LIKE** keyword, since we're using a wildcard to match all East building offices, regardless of number. We'll then put our '**East**' string following the **LIKE** keyword to match any of the East building offices.

Here is the output of this command:

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 (0.043 sec)

## Retrieve Employees in Finance or Sales

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

For this query, we want to see which employees are in Finance or the Sales departments. Since we know about the **SELECT** and **FROM** keywords now, let's go straight to the filter portion of this command.

From the **WHERE** keyword, we'll then search the department column that is equal to the '**Finance**' department. Following this we'll add our **OR** operator so that either or both of these conditions can be met. We'll then use the department column again to search for any employee in the '**Sales**' department.

Here is the output of this command:

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	cggriffin	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
1072	u905v920w694	esmith	Sales	East-421
1076	y347z204a710	fgarcia	Finance	Central-270
1078	a667b270c984	sharley	Sales	North-418
1081	d647e310f618	qcorbit	Finance	South-290
1083	f840g812h544	gkoshi	Finance	West-165

## Retrieve All Employees Not in I.T.

```
MariaDB [organization]> SELECT *
-> FROM employees
-> WHERE NOT department = 'Information Technology';
```

For our last SQL query, we want to retrieve all of the employees who are not in the **'Information Technology'** department. Since we know about the **SELECT** and **FROM** keywords now, let's go straight to the filter portion of this command.



The filter for this query is very straightforward, We'll start with our **WHERE** keyword, and following it we'll add our **NOT** keyword, this will negate the condition for this filter. After that, we'll add the **department** column to then search for specifically the 'Information Technology' department.

Here is the output of this command:

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	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
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
1051	z451a308b518	itraora	Marketing	Central-134

## Summary

In summary, we used SQL commands to analyze data and retrieve specific information from a database. We went over all of the SQL keywords for queries: **SELECT**, **FROM**, **WHERE**, **LIKE**, and **NOT**. As well as understanding wildcards, operators, string, and numeric data types. With this knowledge, we are better equipped to handle SQL as a cybersecurity analyst.