

# Apply filters to SQL queries

## Project description

This scenario project will demonstrate my knowledge and case use of SQL commands and databases.

You are a security professional at a large organization. Part of your job is to investigate security issues to help keep the system secure. You recently discovered some potential security issues that involve login attempts and employee machines.

Your task is to examine the organization's data in their employees and log\_in\_attempts tables. You'll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues.

## Retrieve after hours failed login attempts

You recently discovered a potential security incident that occurred after business hours. To investigate this, you need to query the log\_in\_attempts table and review after hours login activity. Use filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00.

To query data on failed login attempts made after 18:00 in the log\_in\_attempts table, I use this command: `SELECT * FROM log_in_attempts WHERE login_time > "18:00" AND success = 0;`

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

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 (0.298 sec)

MariaDB [organization]>
```

## Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate this event, you want to review all login attempts which occurred on this day and the day before. Use filters in SQL to create a query that identifies all login attempts that occurred on 2022-05-09 or 2022-05-08.

This SQL command will query data on login attempts made on 2022-05-09 or 2022-05-08. The command is: `SELECT * FROM log_in_attempts WHERE login_date = "2022-05-08" OR "2022-05-09";`

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date = "2022-05-08" OR "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
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
6	arutley	2022-05-12	17:00:59	MEXICO	192.168.3.24	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
9	yappiah	2022-05-11	13:47:29	MEX	192.168.59.136	1
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
20	tshah	2022-05-12	18:56:36	MEXICO	192.168.109.50	0
21	iuduike	2022-05-11	17:50:00	US	192.168.131.147	1
22	rjensen	2022-05-11	00:59:26	MEX	192.168.213.128	0
23	yappiah	2022-05-10	18:11:53	MEXICO	192.168.200.48	1
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
27	aalonso	2022-05-10	01:55:35	MEX	192.168.103.210	0
28	aestrada	2022-05-09	19:28:12	MEXICO	192.168.27.57	0
29	bisles	2022-05-11	01:21:22	US	192.168.85.186	0
30	yappiah	2022-05-09	03:22:22	MEX	192.168.124.48	1
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
35	tshah	2022-05-10	15:26:08	MEX	192.168.92.147	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

The SQL query searches through the login\_date column from the log\_in\_attempts based on the criteria after WHERE, it searches for either date.

## Retrieve login attempts outside of Mexico

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, you need to investigate login attempts that occurred outside of Mexico. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico.

For this scenario, instead of retrieving data based on a certain criteria, we can show data based on what not to show. `SELECT * FROM log_in_attempts WHERE NOT country LIKE "MEX%";`

```
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	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	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
44	daquino	2022-05-08	07:02:35	CANADA	192.168.168.144	0
45	dtanaka	2022-05-11	10:28:54	US	192.168.223.157	1
46	eraab	2022-05-11	11:29:27	CAN	192.168.24.12	0
47	dkot	2022-05-08	05:06:45	US	192.168.233.24	1
48	asundara	2022-05-11	03:18:45	USA	192.168.72.10	1
49	asundara	2022-05-08	14:00:01	US	192.168.173.213	0

When we want to query based on something we don't want to see, we use NOT after WHERE to not include the data in the column. We can also use wildcard symbols so that the query auto fills any characters to the right of the symbol, LIKE has to be used when using wildcards instead of “=”.

## Retrieve employees in Marketing

Your team wants to perform security updates on specific employee machines in the Marketing department. You're responsible for getting information on these employee machines and will need to query the employees table. Use filters in SQL to create a query that identifies all employees in the Marketing department for all offices in the East building.

Using SQL, we can query data based on multiple criterias or columns of data. **SELECT \* FROM employees WHERE department = "Marketing" AND office LIKE "East%";**

```

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 | randers | Marketing | East-460 |
| 1156 | a184b775c707 | dellery | Marketing | East-417 |
| 1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.046 sec)

MariaDB [organization]> 

```

This SQL command queries two columns in the employees table.

## Retrieve employees in Finance or Sales

Your team now needs to perform a different security update on machines for employees in the Sales and Finance departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments.

```
SELECT * FROM employees WHERE department = "Sales" OR "Finance";
```

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

employee_id	device_id	username	department	office
1009	NULL	lrodriqu	Sales	South-134
1011	1748m120n401	drosas	Sales	South-292
1024	y976z753a267	iuduike	Sales	South-215
1025	z381a365b233	jhill	Sales	North-115
1035	j236k303l245	bisles	Sales	South-171
1039	n253o917p623	cjackson	Sales	East-378
1041	p929q222r778	cgriffin	Sales	North-208
1057	f370g535h632	mscott	Sales	South-270
1063	l686m140n569	lpope	Sales	East-226
1066	o678p794q957	ttyrell	Sales	Central-444
1071	t244u829v723	zdutchma	Sales	West-348
1072	u905v920w694	esmith	Sales	East-421
1078	a667b270c984	sharley	Sales	North-418
1085	h339i498j269	cperez	Sales	East-325
1086	i281j129k749	lmajumda	Sales	West-499
1089	l358m929n154	jpark2	Sales	West-251
1091	n378o313p469	rtran	Sales	Central-230
1092	o391p779q935	lpark	Sales	West-227
1098	u671v146w618	tarchamb	Sales	North-423
1107	d168e758f876	akajwara	Sales	North-471
1109	f229g533h679	nlocklea	Sales	East-196
1110	g567h376i314	pchaudhu	Sales	Central-428
1111	h835i179j862	jlee	Sales	West-309
1116	m272n572o874	nzhao	Sales	South-100
1117	n683o758p820	dahmad	Sales	West-405
1118	o305p208q337	jpark3	Sales	South-329
1119	p164q780r999	omubarak	Sales	West-409
1121	r628s557t397	mrojas	Sales	East-288
1130	a317b635c465	tsnow	Sales	Central-451
1169	NULL	mmitchel	Sales	Central-250
1176	u849v569w521	nliu	Sales	West-220
1185	d790e839f461	revens	Sales	North-330
1186	e281f433g404	sacosta	Sales	North-460

## Retrieve all employees not in IT

Your 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. Use filters in SQL to create a query which identifies all employees not in the IT department.

```
SELECT * FROM employees WHERE NOT department = "Information Technology";
```

```
MariaDB [organization]> SELECT * FROM employees WHERE NOT department LIKE "Information%";
```

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
1039	n253o917p623	cjackson	Sales	East-378
1040	o783p832q294	dtarly	Human Resources	East-237

In this picture, the command results in the same data as if we were to right it all the way like first shown.

## Summary

In this project, SQL filters are used to analyze login attempts and employee data for security investigations. Key tasks include retrieving failed logins after business hours, identifying logins on targeted dates, and excluding login attempts from Mexico. The project also focuses on extracting employee records for specific departments and locations.