

Apply filters to SQL queries

Project description

I conducted an investigation into employee login activity and workstation update requirements using SQL. The goal was to identify suspicious authentication patterns, review activity around a key date, and determine which employees and devices required targeted updates. Throughout this project, I applied SQL filters using AND, OR, NOT, and LIKE to extract relevant information from large datasets.

Retrieve after hours failed login attempts

```
SELECT *  
FROM log_in_attempts  
WHERE login_time > '18:00'  
AND success = 0;
```

Explanation

This query identifies all failed login attempts that occurred after 18:00. The filter `login_time > '18:00'` selects activity outside business hours, and `success = 0` returns only failed attempts. Using AND ensures both conditions must be true for a record to appear.

```

MariaDB [organization]> clear
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 | 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 |

```

2. Retrieve login attempts on 2022-05-09 or 2022-05-08

```

SELECT *
FROM log_in_attempts
WHERE login_date = '2022-05-09'
OR login_date = '2022-05-08';

```

Explanation

This query returns login attempts from the day of the suspicious event and the day before. Using OR allows the query to match either date, ensuring that all relevant activity is included.

```
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	

3. Retrieve login attempts outside of Mexico

```
SELECT *
FROM log_in_attempts
WHERE NOT country LIKE 'MEX%';
```

Explanation

Some login attempts list the country as "MEX," while others use "MEXICO." The `LIKE 'MEX%'` pattern captures both, and the `NOT` operator reverses the filter to return only login attempts that occurred 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 |

```

4. Retrieve employees in the Marketing department located in East building offices

```

SELECT *
FROM employees
WHERE department = 'Marketing'
AND office LIKE 'East-%';

```

Explanation

This query finds employees who work in the Marketing department and whose offices are in the East building. The **LIKE 'East-%'** pattern matches any East building office regardless of room number, and **AND** ensures both conditions must be satisfied.

```
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 (0.063 sec)
```

```
MariaDB [organization]> 
```

5. Retrieve employees in the Finance or Sales departments

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

Explanation

This query returns all employees whose department is either Finance or Sales. The OR operator allows records that match either department, which is necessary because both require similar security updates.

```
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

6. Retrieve all employees not in the IT department

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

Explanation

This query identifies all employees who are not part of the Information Technology department. The NOT operator excludes those in IT since they have already received the update, leaving only machines that still need the changes.

```
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

SUMMARY

In this project, I used SQL to investigate login activity and employee workstation data as part of a security review. I applied filters using AND, OR, NOT, and LIKE to narrow large datasets into targeted

information necessary for identifying suspicious behavior and determining which employee machines required updates. These queries demonstrate my ability to analyze security data efficiently using SQL.