

Apply filters to SQL queries

Project description

In this scenario, I needed to obtain specific information about employees, their machines, and the departments they belong to from the database.

My team needs data to investigate potential security issues and to update computers.

I was responsible for filtering the required information from the database.

Here's how I performed these tasks: **First**, I retrieved all failed login attempts after business hours. **Second**, I retrieved all login attempts that occurred on specific dates. **Third**, I retrieve logins that didn't originate in Mexico. **Fourth**, I retrieved information about certain employees in the Marketing department. **Fifth**, I retrieved information about employees in the Finance or the Sales department. **Finally**, I obtained information about employees who are not in the Information Technology department.

Retrieve after hours failed login attempts

```
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	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 (0.099 sec)
```

The query aims to find all unsuccessful login attempts made after 18:00. It uses the **AND** operator to combine two conditions:

Failed login attempts: The success column must be 0 (indicating a failed attempt).

After business hours: The login_time must be later than 18:00.

SELECT *: Selects all columns from the table.

FROM log_in_attempts: Specifies the table to query.

WHERE success = 0: Filters for failed login attempts.

AND TIME(login_time) > '18:00': Ensures the login attempts occurred after 18:00.

This query helps your team identify all failed login attempts made after office hours.

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';
```

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

The query retrieves all records from the `log_in_attempts` table where the login attempts were made on either May 9, 2022, or May 8, 2022. It uses the **OR** operator to combine the two conditions.

Here's a breakdown of the query:

SELECT *: This part selects all columns from the table.

FROM `log_in_attempts`: Specifies the table from which to retrieve the data.

WHERE `login_date = '2022-05-09' OR login_date = '2022-05-08'`: Filters the records to include only those where the `login_date` is either '2022-05-09' or '2022-05-08'.

This query helps your team identify login attempts made on these specific dates.

Retrieve login attempts 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	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

The query retrieves all records from the `log_in_attempts` table where the login attempts did not originate from Mexico. It uses the **NOT** operator combined with the **LIKE** operator to exclude entries that start with 'MEX', which covers both 'MEX' and 'MEXICO'.

Here's a breakdown of the query:

SELECT *: This part selects all columns from the table.

FROM `log_in_attempts`: Specifies the table from which to retrieve the data.

WHERE NOT `country` **LIKE** 'MEX%': Filters out records where the country field starts with 'MEX'.

This query helps your team identify login attempts that were made from countries other than Mexico.

Retrieve employees in Marketing

```
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 (0.001 sec)
```

The query retrieves all records from the employees table where the employees are in the 'Marketing' department and are located in any office within the East building.

Here's a breakdown of the query:

SELECT *: This part selects all columns from the table.

FROM employees: Specifies the table from which to retrieve the data.

WHERE department = 'Marketing': Filters the records to include only those where the department is 'Marketing'.

AND office **LIKE** 'East%': Further filters the records to include only those where the office starts with 'East', covering all offices in the East building (e.g., 'East-170', 'East-320').

This query helps your team identify all employees in the Marketing department who are located in the East building.

Retrieve employees in Finance or Sales

```
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
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
1085	h339i498j269	cperez	Sales	East-325

The query retrieves all records from the employees table where the employees are in either the 'Finance' or 'Sales' department. It uses the **OR** operator to include records that meet either of the specified conditions.

Here's a breakdown of the query:

SELECT *: This part selects all columns from the table.

FROM employees: Specifies the table from which to retrieve the data.

WHERE department = 'Finance' **OR** department = 'Sales': Filters the records to include only those where the department is either 'Finance' or 'Sales'.

This query helps your team identify all employees who are in either the Finance or Sales department.

Retrieve all employees not in IT

```
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

The query retrieves all records from the employees table where the employees are not in the 'Information Technology' department. It uses the **NOT** operator to exclude records that match the specified condition.

Here's a breakdown of the query:

SELECT *: This part selects all columns from the table.

FROM employees: Specifies the table from which to retrieve the data.

WHERE NOT department = 'Information Technology': Filters the records to exclude those where the department is 'Information Technology'.

This query helps your team identify all employees who are in departments other than Information Technology.

Summary

Applying filters to SQL queries is a fundamental skill for managing and analyzing data. Filters are used to narrow down the data in a database to only the rows that meet certain criteria, using the WHERE clause. For example, to find customers from a specific city, you would use a query like ``SELECT * FROM customers WHERE city = 'Berlin';``. You can also use comparison operators to filter numerical data, such as finding all orders above a certain amount. Additionally, logical operators like AND, OR, and NOT can be combined to create more complex filters, allowing for precise data extraction and efficient data handling.