

Apply Filters to SQL Queries

Completed by: Hope Tan (github.com/hope-tan)

Project description

In this project, I'm the cybersecurity analyst for a fictional company that stores logs in SQL databases (MariaDB). I have documented the steps I took to make SQL queries that perform security-related tasks to investigate potential risks.

To extract the data I need, I will use operators like "AND", "OR", "NOT", and "LIKE" and wildcards like "%" and "*". Additionally, I will perform an "INNER JOIN", "LEFT JOIN", and "RIGHT JOIN" on two tables to extract relevant data.

Retrieve after hours failed login attempts

To investigate a potential security incident that occurred after business hours, I need to query the "log_in_attempts" table and review after hours login activity.

First, I noticed that the log_in_attempts table has 200 entries.

```
MariaDB [organization]> select * from log_in_attempts;
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	0
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	0
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0

I didn't want to go through all 200 to find the after hours login activity. Therefore, I used the following query to pull all attempts that occurred after 6pm:

```
select * from log_in_attempts where login_time > '18:00' and success = FALSE;
```

```
MariaDB [organization]> select * from log_in_attempts where login_time > '18:00' and success = FALSE;
```

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 (0.056 sec)
```

This returned 19 entries. It is far easier to investigate 19 entries than 200.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. To investigate this event, I will review all login attempts which occurred on this day and the day before using this command:

```
select * from log_in_attempts where login_date = '2022-05-08' or login_date = '2022-05-09';
```

```
MariaDB [organization]> select * from log_in_attempts where login_date = '2022-05-08' or login_date = '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
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

This query returned 75 different entries for logins on May 8th and 9th.

165	jreckley	2022-05-08	15:28:43	MEXICO	192.168.34.193	0
168	jlansky	2022-05-08	13:25:42	USA	192.168.210.94	1
169	alevitsk	2022-05-08	08:10:43	CANADA	192.168.210.228	0
170	sbaelish	2022-05-09	16:43:18	USA	192.168.65.113	0
172	mabadi	2022-05-08	08:06:50	US	192.168.180.41	1
178	sgilmore	2022-05-08	12:27:22	CAN	192.168.52.216	0
184	alevitsk	2022-05-08	03:09:48	CAN	192.168.33.70	0
186	bisles	2022-05-09	04:29:17	USA	192.168.40.72	0
187	arusso	2022-05-09	00:36:26	MEX	192.168.77.137	0
189	nmason	2022-05-08	05:37:24	CANADA	192.168.168.117	1
190	jsoto	2022-05-09	05:09:21	USA	192.168.25.60	0
191	cjackson	2022-05-08	06:46:07	CANADA	192.168.7.187	0
193	lrodriqu	2022-05-08	07:11:29	US	192.168.125.240	0
197	jsoto	2022-05-08	09:05:09	US	192.168.36.21	0

-----+-----+-----+-----+-----+-----+-----+
75 rows in set (0.001 sec)

Retrieve login attempts outside of Mexico

There is suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. I will investigate login attempts that occurred outside of Mexico using filters to exclude that criteria. I used the following command:

```
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

This query returned 144 login attempts from locations outside of Mexico. The table originally had 200 entries, so this cuts the amount of data we need to go through by 56 lines.

191	cjackson	2022-05-08	06:46:07	CANADA	192.168.7.187	0
192	bisles	2022-05-10	08:32:03	USA	192.168.201.40	1
193	lrodriqu	2022-05-08	07:11:29	US	192.168.125.240	0
194	jclark	2022-05-12	14:11:04	CAN	192.168.197.247	0
195	alevitsk	2022-05-11	06:59:13	CANADA	192.168.236.78	1
196	acook	2022-05-10	09:56:48	CAN	192.168.52.90	0
197	jsoto	2022-05-08	09:05:09	US	192.168.36.21	0
200	jclark	2022-05-12	01:11:45	CANADA	192.168.91.103	1

-----+-----+-----+-----+-----+-----+-----+
144 rows in set (0.001 sec)

Retrieve all employees who attempted logging-in

My team needs to investigate the log-in incident further. I will retrieve the information on all employees who have made login attempts with an “INNER JOIN” of the “employees” and “log_in_attempts” tables, linked on the username column. I used the following command:

```
select * from employees inner join log_in_attempts on employees.username = log_in_attempts.username;
```

```
MariaDB [organization]> select * from employees inner join log_in_attempts on employees.username = log_in_attempts.username;
```

employee_id	device_id	username	department	office	event_id	username	login_date	login_time	country	ip_address	success
1032	g773h3031639	jrafael	Information Technology	Central-309	1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	0
1026	a998b568c863	apatel	Human Resources	West-320	2	apatel	2022-05-10	20:27:27	CAN	192.168.205.12	0
1031	f419g188h578	dkot	Marketing	West-408	3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	0
1031	f419g188h578	dkot	Marketing	West-408	4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0
1032	g773h3031639	jrafael	Information Technology	Central-309	5	jrafael	2022-05-11	03:05:59	CANADA	192.168.86.232	0
1020	u899v381w363	arutley	Marketing	South-351	6	arutley	2022-05-12	17:00:59	MEXICO	192.168.3.24	0
1004	e218f877g788	eraab	Human Resources	South-127	7	eraab	2022-05-11	01:45:14	CAN	192.168.170.243	0
1035	j236k3031245	bisles	Sales	South-171	8	bisles	2022-05-08	01:30:17	US	192.168.119.173	0
1033	NULL	yappiah	Information Technology	West-387	9	yappiah	2022-05-11	13:47:29	MEX	192.168.59.136	0

Note that this returned all rows where there was a username in both the employees and log_in_attempts tables.

1009	NULL	lrodrigu	Sales	South-134	193	lrodrigu	2022-05-08	07:11:29	US	192.168.125.240	0
1017	r550s824t230	jclark	Finance	North-188	194	jclark	2022-05-12	14:11:04	CAN	192.168.197.247	0
1006	g329h3571597	alevitsk	Information Technology	East-320	195	alevitsk	2022-05-11	06:59:13	CANADA	192.168.236.78	0
1042	q175r338s833	acook	Human Resources	West-381	196	acook	2022-05-10	09:56:48	CAN	192.168.52.90	0
1015	p611q262r945	jsoto	Finance	North-271	197	jsoto	2022-05-08	09:05:09	US	192.168.36.21	0
1033	NULL	yappiah	Information Technology	West-387	198	yappiah	2022-05-12	10:37:22	MEXICO	192.168.103.106	0
1033	NULL	yappiah	Information Technology	West-387	199	yappiah	2022-05-11	19:34:48	MEXICO	192.168.44.232	0
1017	r550s824t230	jclark	Finance	North-188	200	jclark	2022-05-12	01:11:45	CANADA	192.168.91.103	0

200 rows in set (0.008 sec)

Retrieve employees in Marketing

My team wants to perform security updates on specific employee machines in the Marketing department. I need to query the employees table to identify all employees in the Marketing department for all offices in the East building.

First, I note that the employees table has lots of entries.

```
MariaDB [organization]> select * from employees;
```

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

200 entries, to be exact. It's a lot of work to manually go through 200 employee records to figure out who is in the Marketing department and in an East building.

```

+-----+-----+-----+-----+-----+
| 1196 | o225p357q829 | sshah2 | Information Technology | South-385 |
| 1197 | p791q114r509 | aabara | Information Technology | North-159 |
| 1198 | q308r573s459 | jmartine | Marketing | South-117 |
| 1199 | r520s571t459 | areyes | Human Resources | East-100 |
+-----+-----+-----+-----+-----+
200 rows in set (0.001 sec)

```

Thus, I queried the database with the following command to pull all employees on an East building and in the Marketing department:

```
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 | randerss | Marketing | East-460 |
| 1156 | a184b775c707 | dellery | Marketing | East-417 |
| 1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

```

This means that the department needs to perform updates on the machines of 7 employees. The query made it far easier to identify these 7 employees instead of manually sorting through 200 employee records.

Retrieve employees in Finance or Sales

My team now needs to perform a different security update on machines for employees in the Sales and Finance departments. I created the following query that identifies all employees in the Sales or Finance departments:

```
select * from employees where department = 'Finance' or department = '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 |
+-----+-----+-----+-----+-----+

```

This query returned 71 employees from either Sales or Finance, so the department needs to update the machines of these 71 employees. Again, the query made it easier to directly identify these employees, so I didn't have to manually sort 200 records to determine who needed this update.

1159	d881e710f732	jshen	Finance	East-193
1164	i682j513k442	fsmeltz	Finance	North-163
1169	NULL	mmitchel	Sales	Central-250
1174	s371t911u987	eortiz	Finance	North-428
1175	t959u687v394	jclark2	Finance	North-194
1176	u849v569w521	nliu	Sales	West-220
1181	z803a233b718	sessa	Finance	South-207
1185	d790e839f461	revens	Sales	North-330
1186	e281f433g404	sacosta	Sales	North-460
1187	f963g637h851	bbode	Finance	East-351
1188	g164h566i795	noshiro	Finance	West-252
1195	n516o853p957	orainier	Finance	East-346

71 rows in set (0.001 sec)

Retrieve all employees not in IT

My 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. I will identify all employees not in the IT department using SQL filters.

select * from employees where not department = 'Information Technology';

```
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

This query returned 161 employees outside of the Information Technology department.

1186	e281f433g404	sacosta	Sales	North-460
1187	f963g637h851	bbode	Finance	East-351
1188	g164h566i795	noshiro	Finance	West-252
1189	h784i120j837	slefkowi	Human Resources	West-342
1190	NULL	kcarter	Marketing	Central-270
1191	NULL	shakimi	Marketing	Central-366
1194	m340n287o441	zwarren	Human Resources	West-212
1195	n516o853p957	orainier	Finance	East-346
1198	q308r573s459	jmartine	Marketing	South-117
1199	r520s571t459	areyes	Human Resources	East-100

161 rows in set (0.001 sec)

Perform inner join for “employees” and “machines” tables

I want to identify which employees are using which machines. To retrieve this information, I will “INNER JOIN” the “employees” table (with employee information) to the “machines” table (with machine information).

The employees table has 200 entries and includes fields for employee_id, device_id, username, department, and office.

```
MariaDB [organization]> select * from employees;
```

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

The machines table has 200 entries for devices and includes fields for device_id, operating_system, email_client, OS_patch_date, and employee_id.

```
MariaDB [organization]> select * from machines;
```

device_id	operating_system	email_client	OS_patch_date	employee_id
a184b775c707	OS 1	Email Client 1	2021-09-01	1156
a192b174c940	OS 2	Email Client 1	2021-06-01	1052
a305b818c708	OS 3	Email Client 2	2021-06-01	1182
a317b635c465	OS 1	Email Client 2	2021-03-01	1130
a320b137c219	OS 2	Email Client 2	2021-03-01	1000

I note that these tables share the “employee_id” field. Therefore, I want to base my INNER JOIN on that employee_id field. I entered the following command:

```
select * from machines inner join employees on machines.device_id = employees.device_id;
```

```
MariaDB [organization]> select * from machines inner join employees on machines.device_id = employees.device_id;
```

device_id	operating_system	email_client	OS_patch_date	employee_id	employee_id	device_id	username	department	office
a320b137c219	OS 2	Email Client 2	2021-03-01	1000	1000	a320b137c219	elarson	Marketing	East-170
b239c825d303	OS 1	Email Client 1	2021-03-01	1001	1001	b239c825d303	bmoreno	Marketing	Central-276
c116d593e558	OS 3	Email Client 1	2021-09-01	1002	1002	c116d593e558	tshah	Human Resources	North-434
d394e816f943	OS 3	Email Client 2	2021-03-01	1003	1003	d394e816f943	sgilmore	Finance	South-153
e218f877g788	OS 2	Email Client 1	2021-09-01	1004	1004	e218f877g788	eraab	Human Resources	South-127
f551g340h864	OS 3	Email Client 2	2021-12-01	1005	1005	f551g340h864	gesparza	Human Resources	South-366
g329h357i597	OS 1	Email Client 2	2021-06-01	1006	1006	g329h357i597	alevitsk	Information Technology	East-320
h174i497j413	OS 2	Email Client 1	2021-03-01	1007	1007	h174i497j413	wjaffrey	Finance	North-406

This returned 185 entries for employees with machines. As a result, I could quickly identify which employee owned which machine.

```

1186m618n319 | OS 1 | Email Client 2 | 2021-12-01 | 1193 | 1193 | 1186m618n319 | esantiago | Information Technology | Central-30
0 |
m340n287o441 | OS 2 | Email Client 2 | 2021-09-01 | 1194 | 1194 | m340n287o441 | zwarren | Human Resources | West-212
n516o853p957 | OS 1 | Email Client 1 | 2021-09-01 | 1195 | 1195 | n516o853p957 | orainier | Finance | East-346
o225p357q829 | OS 3 | Email Client 1 | 2021-12-01 | 1196 | 1196 | o225p357q829 | sshah2 | Information Technology | South-385
p791q114r509 | OS 2 | Email Client 1 | 2021-09-01 | 1197 | 1197 | p791q114r509 | aabara | Information Technology | North-159
q308r573s459 | OS 3 | Email Client 1 | 2021-03-01 | 1198 | 1198 | q308r573s459 | jmartine | Marketing | South-117
r520s571t459 | OS 2 | Email Client 2 | 2021-03-01 | 1199 | 1199 | r520s571t459 | areyes | Human Resources | East-100
---+
185 rows in set (0.002 sec)

```

Perform left join for “employees” and “machines” tables

I want to retrieve data on all machines and any employees who own a machine. To retrieve this information, I will “LEFT JOIN” the machines table to the employees table using the following command:

```
select * from machines left join employees on machines.device_id = employees.device_id;
```

```

MariaDB [organization]> select * from machines left join employees on machines.device_id = employees.device_id;

```

device_id	operating_system	email_client	OS_patch_date	employee_id	employee_id	device_id	username	department	office
a320b137c219	OS 2	Email Client 2	2021-03-01	1000	1000	a320b137c219	elarson	Marketing	East-170
b239c825d303	OS 1	Email Client 1	2021-03-01	1001	1001	b239c825d303	bmoreno	Marketing	Central-276
c116d593e558	OS 3	Email Client 1	2021-09-01	1002	1002	c116d593e558	tshah	Human Resources	North-434
d394e816f943	OS 3	Email Client 2	2021-03-01	1003	1003	d394e816f943	sgilmore	Finance	South-153

Notice some NULL values for entries from the employees table, representing machines that do not have an employee assigned to them.

```

v846w200x439 | OS 1 | Email Client 1 | 2021-06-01 | 0 | NULL | NULL | NULL | NULL | NULL |
w981x771y326 | OS 2 | Email Client 2 | 2021-06-01 | 0 | NULL | NULL | NULL | NULL | NULL |
x561y839z458 | OS 2 | Email Client 1 | 2021-09-01 | 0 | NULL | NULL | NULL | NULL | NULL |
y246z508a775 | OS 2 | Email Client 1 | 2021-12-01 | 0 | NULL | NULL | NULL | NULL | NULL |
z821a946b264 | OS 3 | Email Client 2 | 2021-06-01 | 0 | NULL | NULL | NULL | NULL | NULL |
---+
200 rows in set (0.008 sec)

```

Perform right join for “employees” and “machines” tables

I want to retrieve data on all employees and any machines they use. To retrieve this information, I will “RIGHT JOIN” the machines table to the employees table using the following command:

```
select * from machines right join employees on machines.device_id = employees.device_id;
```

```

MariaDB [organization]> select * from machines right join employees on machines.device_id = employees.device_id;

```

device_id	operating_system	email_client	OS_patch_date	employee_id	employee_id	device_id	username	department	office
a320b137c219	OS 2	Email Client 2	2021-03-01	1000	1000	a320b137c219	elarson	Marketing	East-170
b239c825d303	OS 1	Email Client 1	2021-03-01	1001	1001	b239c825d303	bmoreno	Marketing	Central-276
c116d593e558	OS 3	Email Client 1	2021-09-01	1002	1002	c116d593e558	tshah	Human Resources	North-434
d394e816f943	OS 3	Email Client 2	2021-03-01	1003	1003	d394e816f943	sgilmore	Finance	South-153
e218f877g788	OS 2	Email Client 1	2021-09-01	1004	1004	e218f877g788	eraab	Human Resources	South-127
f551g340h864	OS 3	Email Client 2	2021-12-01	1005	1005	f551g340h864	gesparza	Human Resources	South-366
g329h357i597	OS 1	Email Client 2	2021-06-01	1006	1006	g329h357i597	alevitsk	Information Technology	East-320
h174i497j413	OS 2	Email Client 1	2021-03-01	1007	1007	h174i497j413	wjaffrey	Finance	North-406
i858j583k571	OS 2	Email Client 2	2021-06-01	1008	1008	i858j583k571	abernard	Finance	South-170
NULL	NULL	NULL	NULL	1009	1009	NULL	lrodriqu	Sales	South-134

Note some NULL values in some rows that came from the machines table. This represents employees who don't have a machine assigned to them.

NULL	NULL	NULL	NULL	NULL	1190	NULL	kcarter	Marketing	Central-270
NULL	NULL	NULL	NULL	NULL	1191	NULL	shakimi	Marketing	Central-366
k5701183m949	OS 3	Email Client 1	2021-12-01	1192	1192	k5701183m949	rlaghari	Information Technology	East-138
l186m618n319	OS 1	Email Client 2	2021-12-01	1193	1193	l186m618n319	esantiago	Information Technology	Central-300
m340n287o441	OS 2	Email Client 2	2021-09-01	1194	1194	m340n287o441	zwarren	Human Resources	West-212
n516o853p957	OS 1	Email Client 1	2021-09-01	1195	1195	n516o853p957	orainier	Finance	East-346
o225p357q829	OS 3	Email Client 1	2021-12-01	1196	1196	o225p357q829	sshah2	Information Technology	South-385
p791q114r509	OS 2	Email Client 1	2021-09-01	1197	1197	p791q114r509	aabara	Information Technology	North-159
q308r573s459	OS 3	Email Client 1	2021-03-01	1198	1198	q308r573s459	jmartine	Marketing	South-117
r520s571t459	OS 2	Email Client 2	2021-03-01	1199	1199	r520s571t459	areyes	Human Resources	East-100

200 rows in set (0.001 sec)

Summary

In this project, I leveraged my knowledge of SQL to perform security tasks. I queried for information from the “log_in_attempts” and “employees” tables using operators like “AND”, “OR”, “NOT”, and “LIKE” and wildcards like “%” and “*” to filter the outputs to extract the exact data I needed. I performed an “INNER JOIN”, “LEFT JOIN”, and “RIGHT JOIN” on two tables.