# Using SQL joins to obtain the precise data

## **Project description**

The leadership team at my organization tasked me with assessing potential security risks and implementing necessary updates on employee computers. As a Linux administrator, I utilized SQL queries with filters and JOIN techniques to perform security-related tasks. These efforts involved analysing and integrating data from two interconnected tables, demonstrating my expertise in relational database management and security analysis.

## **Match Employees to their Machines**

To link employees with their assigned machines, I wrote an SQL query on MariaDB to join two tables: machines and employees. The query focused on the intersection of both tables, ensuring that only rows with matching values were included.

```
MariaDB [organization] > SELECT *

->
-> FROM employee;

| employee_id | device_id | username | department | office |

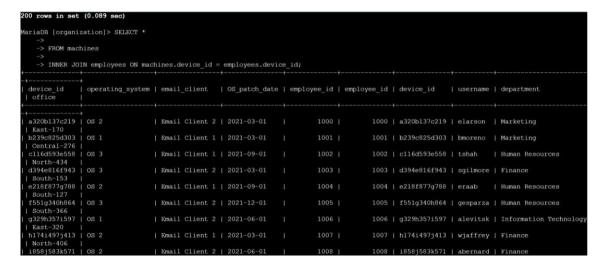
| 1000 | a320b137c219 | elarson | Marketing | East-170 |
| 1001 | b239c8254303 | bmoreno | Marketing | Central-276 |
| 1002 | cl1cd6939c558 | tshah | Human Resources | North-434 |
| 1003 | d394c816f943 | sqilmore | Finance | South-153 |
| 1004 | e218f877g788 | eraab | Human Resources | South-127 |
| 1005 | f55ig340h864 | gesparza | Human Resources | South-127 |
| 1006 | d329h357i1997 | alevitsk | Information Technology | East-320 |
| 1007 | h174f497j413 | wjaffrey | Finance | South-170 |
| 1008 | 1858j583k571 | abernard | Finance | South-104 |
| 1008 | 1858j583k571 | abernard | Finance | South-104 |
| 1010 | k242121zm542 | jlansky | Finance | South-104 |
| 1011 | 174@m120n401 | drosas | Sales | South-109 |
| 1012 | m756n6680146 | mmason | Information Technology | North-160 |
| 1013 | n2050559p243 | zbernal | Information Technology | North-271 |
| 1016 | q79373765288 | sbaelish | Human Resources | North-271 |
| 1017 | r550824t230 | jclark | Finance | North-188 |
| 1018 | s310t5400653 | abelims | Finance | North-108 |
| 1019 | t815u205v470 | mcouliba | Information Technology | North-108 |
| 1021 | v200u121x977 | martell | Information Technology | North-108 |
| 1021 | v200u121x977 | martell | Information Technology | North-108 |
| 1021 | v20v121x977 | martell | Information Technology | North-108 |
| 1021 | v20v121x977 | martell | Information Technology | North-108 |
| 1021 | v20v121x977 | martell | Information Technology | North-108 |
| 1021 | v20v121x977 | martell | Information Technology | North-108 |
| 1022 | v237x430y5c7 | arusso | Finance | West-405 |
| 1023 | v257x592103 | alonso | Information Technology | West-393 |
```

The SQL query retrieved the following data:

- Username (from the relevant table)
- Operating system (from the relevant table)
- Employee ID (from the employees table)

To avoid ambiguity for shared column names, I used the table.column format.

The result included 185 rows, each representing a unique username, operating system, and device ID, providing a comprehensive view of the assigned machines.

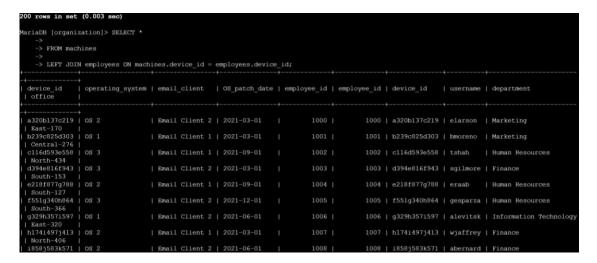


Central-247									
d790e839f461	OS 1	Email	Client 1	2021-06-01	1185	1185	d790e839f461	revens	Sales
North-330									
e281f433g404	os 1	Email	Client 2	2021-12-01	1186	1186	e281f433g404	sacosta	Sales
North-460									
f963g637h851	os 1	Email	Client 1	2021-06-01	1187	1187	f963g637h851	bbode	Finance
East-351   g164h566i795	00.1	1 Percent 1	23 Court 2 1	2021-09-01	1188	1100 1	g164h566i795		l Pinner
G164N5661/95   West-252	05 1	Email	Client 1	2021-09-01	1100	1100	g164n5661/95	nosniro	Finance
h784i120i837	ng 3	I Empil	Client 2	2021-12-01	1189	1189	h784i120i837	elefkowi	Human Resources
West-342		1 miner	oriene s 1	EGET TE OT			HIOTELEOJOSI	DIGINOWI	Hallan Resources
k5701183m949	os 3	Email	Client 1	2021-12-01	1192	1192	k5701183m949	rlaghari	Information Technology
East-138									
1186m618n319	os 1	Email	Client 2	2021-12-01	1193	1193	1186m618n319	esantiag	Information Technology
Central-300									
m340n287o441	OS 2	Email	Client 2	2021-09-01	1194	1194	m340n287o441	zwarren	Human Resources
West-212									
n5160853p957	os 1	Email	Client 1	2021-09-01	1195	1195	n5160853p957	orainier	Finance
East-346									
o225p357q829	OS 3	Email	Client 1	2021-12-01	1196	1196	o225p357q829	sshah2	Information Technology
South-385   p791g114r509	00.0	1 Personal	Client 1	2021 00 01	1197	1107 1	-701-114-E00	- make me	Information Technology
North-159	05 2	Email	Client 1	2021-09-01	119/	119/	b/aidii4i20a	aabara	Information Technology
q308r573s459	08.3	1 Email	Client 1 1	2021-03-01	1198	1198 1	q308r573s459	imartine	Marketing
South-117		Latter 1	and the same of	NOUN ON OR			400000000000000000	James Garre	1 marke saring
r520s571t459	OS 2	Email	Client 2	2021-03-01	1199	1199	r520s571t459	areyes	Human Resources
East-100									
Edst-100 									
185 rows in set	(0.100 res)								
.oJ rows in sec	(0.100 500)								

### Return more data

#### **LEFT JOIN**:

- Ensured that all rows from the machines table were included, regardless of whether they had matching entries in the employees table.
- Used the device id column as the link between the tables.
- Highlighted all machines, including those not assigned to an employee, by populating unmatched rows from the employees table with NULL values.



#### 1. RIGHT JOIN:

o Ensured that all rows from the employees table were included, even if they lacked matching entries in the machines table.

o Highlighted all employees, including those without assigned machines, by populating unmatched rows from the machines table with NULL values.

Both **JOIN** queries produced 200 rows each. However, some columns contained NULL values, reflecting the absence of corresponding data in the related table.

## Retrieve login attempt data

To further investigate the security incident, I retrieved details of all employees who had made login attempts. This involved performing an **INNER JOIN** on the employees and log\_in\_attempts tables, using the username column as the link. This query provided a focused dataset of relevant login activities, supporting a deeper analysis of potential security risks.

### **Summary**

I developed SQL queries to join two tables, analysing three specific scenarios:

- 1. **Inner Join**: Retrieved only the rows with matching values in both tables, focusing on shared data.
- 2. **Left Join**: Included all rows from the machines table, with unmatched employees data represented as NULL.
- 3. **Right Join**: Included all rows from the employees table, with unmatched machines data represented as NULL.

Each query was tailored to explore different aspects of the data, providing critical insights for relational database management and security analysis. The investigation of login attempts via **INNER JOIN** further supported security risk assessments by linking employees to their activity logs.