

# Apply SQL JOIN

## Project description

The management at my organization has asked me to investigate potential security issues and update employee computers as required. As a Linux administrator, I used SQL with filters to perform security-related tasks. But for this scenario, I applied SQL join to play with the values of the two tables. This task is related to relational database management.

## Inner Join

I created a SQL query on MariaDB to join two tables focusing on the intersection of two tables, where we only care about rows that have corresponding values in both.

This is the “employees” table.

```
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
1005	f551g340h864	gesparza	Human Resources	South-366
1006	g329h357i597	alevitsk	Information Technology	East-320
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170
1009	NULL	lrodrigu	Sales	South-134
1010	k242l212m542	jlansky	Finance	South-109
1011	l748m120n401	drozas	Sales	South-292
1012	m756n668o146	nmason	Information Technology	North-160
1013	n205o559p243	zbernal	Information Technology	South-229
1014	NULL	asundara	Information Technology	West-219
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
1019	t815u205v470	mcouliba	Information Technology	North-108
1020	u899v381w363	arutley	Marketing	South-351
1021	v200w121x977	swartell	Information Technology	South-138
1022	w237x430y567	arusso	Finance	West-465
1023	x253y759z103	aalonso	Information Technology	West-393
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	aestrada	Human Resources	West-121
1029	d336e475f676	ivelasco	Finance	East-156
1030	e391f189g913	mabadi	Marketing	West-375
1031	f419g188h578	dkot	Marketing	West-408
1032	g773h303i639	jrafael	Information Technology	Central-309
1033	NULL	yappiah	Information Technology	West-387
1034	i679j565k940	bsand	Human Resources	East-484
1035	j236k303l245	bisles	Sales	South-171

This is the “machines” table.

```
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
a398b471c573	OS 3	Email Client 2	2021-12-01	0
a667b270c984	OS 1	Email Client 1	2021-03-01	1078
a821b452c176	OS 2	Email Client 2	2021-12-01	1104
a998b568c863	OS 3	Email Client 1	2021-12-01	1026
b157c491d493	OS 2	Email Client 1	2021-03-01	0
b239c825d303	OS 1	Email Client 1	2021-03-01	1001
b264c773d977	OS 2	Email Client 2	2021-03-01	1157
b265c937d713	OS 2	Email Client 1	2021-09-01	1131
b433c245d868	OS 1	Email Client 1	2021-06-01	1079
b551c837d758	OS 3	Email Client 1	2021-03-01	1105
b566c710d544	OS 1	Email Client 1	2021-06-01	1183
b806c503d354	OS 2	Email Client 1	2021-12-01	1027
b979c871d361	OS 2	Email Client 1	2021-03-01	1053
c116d593e558	OS 3	Email Client 1	2021-09-01	1002
c150d982e144	OS 2	Email Client 2	2021-06-01	1132
c185d679e493	OS 1	Email Client 2	2021-09-01	0
c406d877e950	OS 2	Email Client 1	2021-06-01	1158
c547d140e477	OS 2	Email Client 1	2021-03-01	1054
c568d742e974	OS 2	Email Client 2	2021-09-01	1080
c597d792e215	OS 2	Email Client 1	2021-09-01	1106
c603d749e374	OS 1	Email Client 1	2021-12-01	1028
c986d200e170	OS 2	Email Client 2	2021-09-01	1184
d168e758f876	OS 2	Email Client 1	2021-09-01	1107
d280e557f635	OS 3	Email Client 1	2021-03-01	0
d336e475f676	OS 2	Email Client 2	2021-06-01	1029
d394e816f943	OS 3	Email Client 2	2021-03-01	1003
d647e310f618	OS 2	Email Client 2	2021-06-01	1081
d693e351f221	OS 2	Email Client 2	2021-09-01	1133
d790e839f461	OS 1	Email Client 1	2021-06-01	1185
d831e972f553	OS 1	Email Client 1	2021-09-01	1055
d881e710f732	OS 3	Email Client 2	2021-03-01	1159
e113f288g203	OS 2	Email Client 2	2021-03-01	1108

This is the query that produces username, operating system, and employee ID from both tables. The username is from one of the tables and so is the operating system. For the table that can be found on both, we use “table.column” format to avoid ambiguity. In this case it is the employee table (employee ID). As the result goes, there are 185 usernames with respective operating systems and device IDs. Other variations can be found in the next pages of Inner Join.

```
MariaDB [organization]> SELECT username, operating_system, employees.device_id
->
->
->
-> FROM employees
->
->
->
-> INNER JOIN machines ON employees.device_id = machines.device_id;
```

username	operating_system	device_id
el Larson	OS 2	a320b137c219
bmoreno	OS 1	b239c825d303
tshah	OS 3	c116d593e558
sgilmore	OS 3	d394e816f943
eraab	OS 2	e218f877g788
gesparza	OS 3	f551g340h864
alevitsk	OS 1	g329h357i597
wjaffrey	OS 2	h174i497j413
abernard	OS 2	i858j583k571
jlansky	OS 1	k242l212m542
drosas	OS 3	l748m120n401
nmason	OS 1	m756n668o146
zbernal	OS 1	n205o559p243
jsoto	OS 1	p611q262r945
sbaelish	OS 2	q793r736s288
jclark	OS 2	r550s824t230
abellmas	OS 1	s310t540u653
mcouliba	OS 3	t815u205v470
arutley	OS 3	u899v381w363
smartell	OS 2	v200w121x977
arusso	OS 2	w237x430y567
aalonso	OS 1	x253y759z103

lquraish	OS 1	b566c710d544
ptsosie	OS 2	c986d200e170
revens	OS 1	d790e839f461
sacosta	OS 1	e281f433g404
hbode	OS 1	f963g637h851
noshiro	OS 1	g164h566i795
slefkowi	OS 3	h784i120j837
rlaghari	OS 3	k570l183m949
esantiag	OS 1	l186m618n319
zwarren	OS 2	m340n287o441
orainier	OS 1	n516o853p957
sshah2	OS 3	o225p357q829
aabara	OS 2	p791q114r509
jmartine	OS 3	q308r573s459
areyes	OS 2	r520s571t459

185 rows in set (0.001 sec)

This query will produce username, employee ID, operating system, device ID and their respective office.

```
MariaDB [organization]> SELECT username, employees.employee_id, operating_system, employees.device_id, office FROM employees INNER JOIN machines ON employees.device_id = machines.device_id;
```

username	employee_id	operating_system	device_id	email_client	office
el Larson	1000	OS 2	a320b137c219	Email Client 2	East-170
bmoreno	1001	OS 1	b239c825d303	Email Client 1	Central-276
tshah	1002	OS 3	c116d593e558	Email Client 1	North-434
sgilmore	1003	OS 3	d394e816f943	Email Client 2	South-153
eraab	1004	OS 2	e218f877g788	Email Client 1	South-127
gesparza	1005	OS 3	f551g340h864	Email Client 2	South-366
alevitsk	1006	OS 1	g329h357i597	Email Client 2	East-320
wjaffrey	1007	OS 2	h174i497j413	Email Client 1	North-406
abernard	1008	OS 2	i858j583k571	Email Client 2	South-170
jlansky	1010	OS 1	k242l212m542	Email Client 1	South-109
drosas	1011	OS 3	l748m120n401	Email Client 1	South-292
nmason	1012	OS 1	m756n668o146	Email Client 2	North-160
sbernal	1013	OS 1	n205o559p243	Email Client 2	South-229
jsoto	1015	OS 1	p611q262r945	Email Client 2	North-271
sbaelish	1016	OS 2	q793r736s288	Email Client 1	North-229
jclark	1017	OS 2	r550s824t230	Email Client 1	North-188
abellmas	1018	OS 1	s310t540u653	Email Client 2	North-403
mcouliba	1019	OS 3	t815u205v470	Email Client 1	North-108
arutley	1020	OS 3	u899v381w363	Email Client 1	South-351
smartell	1021	OS 2	v200w121x977	Email Client 2	South-138
arusso	1022	OS 2	w237x430y567	Email Client 2	West-465
aalonso	1023	OS 1	x253y759z103	Email Client 2	West-393
iuduike	1024	OS 2	y976z753a267	Email Client 2	South-215
jhill	1025	OS 3	z381a365b233	Email Client 2	North-115
apatel	1026	OS 3	a998b568c863	Email Client 1	West-320
mrah	1027	OS 2	b806c503d354	Email Client 1	West-246
aestrada	1028	OS 1	c603d749e374	Email Client 1	West-121
ivelasco	1029	OS 2	d336e475f676	Email Client 2	East-156
mabadi	1030	OS 3	e391f189g913	Email Client 2	West-375
dkot	1031	OS 1	f419g188h578	Email Client 1	West-408
jrafael	1032	OS 2	g773h303i639	Email Client 2	Central-309
bsand	1034	OS 1	i679j565k940	Email Client 1	East-484
bisles	1035	OS 1	j236k303l245	Email Client 1	South-171
rjensen	1036	OS 3	k550l533m205	Email Client 2	Central-239
dtanaka	1037	OS 3	l693m585n528	Email Client 1	West-468
btang	1038	OS 1	m873n636o225	Email Client 2	Central-260
cjackson	1039	OS 1	n253o917p623	Email Client 2	East-378
dtarlv	1040	OS 2	o783p832q294	Email Client 2	East-237

aezra	1177	OS 1	w691w183x928	Email Client 2	East-190
nlannist	1178	OS 3	w986x187y885	Email Client 2	North-196
asalas	1179	OS 1	x174y934z376	Email Client 2	North-445
medwards	1180	OS 2	y131z211a578	Email Client 1	Central-340
sessa	1181	OS 1	z803a233b718	Email Client 2	South-207
mmora	1182	OS 3	a305b818c708	Email Client 2	Central-250
lquraish	1183	OS 1	b566c710d544	Email Client 1	East-400
ptsosie	1184	OS 2	c986d200e170	Email Client 2	Central-247
revens	1185	OS 1	d790e839f461	Email Client 1	North-330
sacosta	1186	OS 1	e281f433g404	Email Client 2	North-460
bbode	1187	OS 1	f963g637h851	Email Client 1	East-351
noshiro	1188	OS 1	g164h566i795	Email Client 1	West-252
sleftowi	1189	OS 3	h784i120j837	Email Client 2	West-342
rlaghari	1192	OS 3	k570l183m949	Email Client 1	East-138
esantiag	1193	OS 1	l186m618n319	Email Client 2	Central-300
zwarren	1194	OS 2	m340n287o441	Email Client 2	West-212
orainier	1195	OS 1	n516o853p957	Email Client 1	East-346
sshah2	1196	OS 3	o225p357q829	Email Client 1	South-385
aabara	1197	OS 2	p791q114r509	Email Client 1	North-159
jmartine	1198	OS 3	q308r573s459	Email Client 1	South-117
areyes	1199	OS 2	s520s571t459	Email Client 2	East-100

185 rows in set (0.001 sec)

## Return More Data

Left Join. The results will include all records from one or the other table. Here, I have to link these tables using the common `device_id` column. In a left join, all records after `FROM` and before `LEFT JOIN` are included in the result. In this case, all records from the `machines` table are included, whether they are assigned to the `employees` table or not.

```

MariaDB [organization]> clear
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-17
b239c825d303	OS 1	Email Client 1	2021-03-01	1001	1001	b239c825d303	bmoreno	Marketing	Central
c116d593e558	OS 3	Email Client 1	2021-09-01	1002	1002	c116d593e558	tshah	Human Resources	North-4
d394e816f943	OS 3	Email Client 2	2021-03-01	1003	1003	d394e816f943	sgilmore	Finance	South-1
e218f877g788	OS 2	Email Client 1	2021-09-01	1004	1004	e218f877g788	eraab	Human Resources	South-1
f551g340h864	OS 3	Email Client 2	2021-12-01	1005	1005	f551g340h864	gesparra	Human Resources	South-3
g329h357i597	OS 1	Email Client 2	2021-06-01	1006	1006	g329h357i597	alevitsk	Information Technology	East-32
h174i497j413	OS 2	Email Client 1	2021-03-01	1007	1007	h174i497j413	wjaffrey	Finance	North-4
i858j583k571	OS 2	Email Client 2	2021-06-01	1008	1008	i858j583k571	abernard	Finance	South-1
k242l212m542	OS 1	Email Client 1	2021-03-01	1010	1010	k242l212m542	jlansky	Finance	South-1

Right Join.

Right Join. The results will include all records from one or the other table. Here, I have to link these tables using the common `device_id` column. In a right join, all records after `RIGHT JOIN` are included in the result. In this case, all records from the `employees` table are

included, whether they have values on the `machine` table or not.

```
mariaDB [organisation]> 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-17
b239c825d303	OS 1	Email Client 1	2021-03-01	1001	1001	b239c825d303	bmoreno	Marketing	Central
c116d593e558	OS 3	Email Client 1	2021-09-01	1002	1002	c116d593e558	tsah	Human Resources	North-4
d394e816f943	OS 3	Email Client 2	2021-03-01	1003	1003	d394e816f943	sgilmore	Finance	South-1
e218f877g788	OS 2	Email Client 1	2021-09-01	1004	1004	e218f877g788	eraab	Human Resources	South-1
f551g340h864	OS 3	Email Client 2	2021-12-01	1005	1005	f551g340h864	gesparza	Human Resources	South-3
g329h357i597	OS 1	Email Client 2	2021-06-01	1006	1006	g329h357i597	alevitsk	Information Technology	East-32
h174i497j413	OS 2	Email Client 1	2021-03-01	1007	1007	h174i497j413	wjaffrey	Finance	North-4
i858j583k571	OS 2	Email Client 2	2021-06-01	1008	1008	i858j583k571	abernard	Finance	South-1
NULL	NULL	NULL	NULL	NULL	1009	NULL	lrodriqu	Sales	South-1
k242l212m542	OS 1	Email Client 1	2021-03-01	1010	1010	k242l212m542	jlansky	Finance	South-1

Both produced 200 rows each, however in the process, some data are written `NUL` due to types of `JOIN`.

## Summary

I wrote queries to join two tables in three different scenarios: Inner Join, Left Join, and Right Join.

## Bonus Part

### Left Join

Scenario: An online store wants to generate a report of all orders placed, along with customer details if available, but also including those without associated customer accounts.

**Orders Table:**

OrderID	CustomerID	OrderDate	OtherOrderDetails
-----	-----	-----	-----
10308	1	2023-12-20	...
10309	2	2023-12-21	...
10310	NULL	2023-12-22	...

**Customers Table:**

CustomerID	CustomerName	Email	OtherCustomerDetails
-----	-----	-----	-----
1	John Smith	jsmith@example.com	...
2	Jane Doe	jane.doe@example.com	...

Query:

SQL

```
SELECT Orders.OrderID, Orders.OrderDate, Customers.CustomerName, Customers.Email
FROM Orders
LEFT JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

Output:

OrderID	OrderDate	CustomerName	Email
10308	2023-12-20	John Smith	jsmith@example.com
10309	2023-12-21	Jane Doe	jane.doe@example.com
10310	2023-12-22	NULL	NULL

Left Join will return all rows from the left table (Orders) even if there's no match in the right table (Customers). For the values that can not be found, it will be shown as NULL.

## Right Join

Scenario: A music streaming service wants to identify customers who haven't listened to any songs in the past week, along with their subscription details.

Customers Table:

CustomerID	SubscriptionType	OtherCustomerDetails
456	Premium	...
789	Free	...

SongsPlayed Table:

CustomerID	SongID	PlayDate	OtherSongDetails
456	12345	2023-12-25	...

Query:

SQL

```
SELECT Customers.CustomerID, Customers.SubscriptionType, SongsPlayed.SongID, SongsPlayed.PlayDate
FROM Customers
RIGHT JOIN SongsPlayed ON Customers.CustomerID = SongsPlayed.CustomerID
WHERE SongsPlayed.PlayDate >= DATE_SUB(CURDATE(), INTERVAL 7 DAY);
```



**Output:**

CustomerID	SubscriptionType	SongID	PlayDate
456	Premium	12345	2023-12-25
789	Free	NULL	NULL

Right Join will return all rows from the Right table (SongsPlayed) even if there's no match in the left table (Customers). For the values that can not be found, it will be shown as NULL.