

Activity: Complete a join

Introduction

In this lab, you'll use `INNER JOIN`, `LEFT JOIN`, and `RIGHT JOIN` in SQL to retrieve information from two different tables. You'll use these different types of SQL joins to join data from separate machines, employees, and login attempts tables. You'll be using the MariaDB shell to run SQL queries.

What you'll do

You have multiple tasks in this lab:

- Use an inner join to find information on employees and their machines
- Use a left join and right join to find information on employees and their machines
- Use an inner join to find information on employees and their login attempts

Activity: Complete a SQL join

1 hourFree

Activity overview

As a security analyst, you'll often find that you need data from more than one table.

Previously, you learned that a **relational database** is a structured database containing tables that are related to each other.

SQL joins enable you to combine tables that contain a shared column. This is helpful when you need to connect information that appears in different tables.

In this lab activity, you'll use SQL joins to connect separate tables and retrieve needed information.

Get ready to apply what you've learned and **join** some data!

Note: The terms **row** and **record** are used interchangeably.

Scenario

In this scenario, you'll investigate a recent security incident that compromised some machines.

You are responsible for getting the required information from the database for the investigation.

Here's how you'll do this task: **First**, you'll use an inner join to identify which employees are using which machines. **Second**, you'll use left and right joins to find machines that do not belong to any specific user and users who do not have any specific machine

assigned to them. **Finally**, you'll use an inner join to list all login attempts made by all employees.

You're ready to join tables in SQL!

Note: *In this lab you'll be working with the organization database and the tables it contains.*

The lab starts with the organization database in the MariaDB shell that is already open.

*This means you can start with the tasks as soon as you click the **Start Lab** button.*

If you unintentionally exit the organization database in the MariaDB shell, you can reconnect by running the `sudo mysql organization` command.

Start your lab

Before you begin, you can review the instructions for using the Qwiklabs platform under the **Resources** tab in Coursera.

If you haven't already done so, click **Start Lab**. This brings up the terminal so that you can begin completing the tasks!

When you have completed all the tasks, refer to the **End your Lab** section that follows the tasks for information on how to end your lab.

Task 1. Match employees to their machines

First, you must identify which employees are using which machines. The data is located in the `machines` and `employees` tables.

You must use a SQL inner join to return the records you need based on a connecting column. In the scenario, both tables include the `device_id` column, which you'll use to perform the join.

1. Run the following query to retrieve all records from the `machines` table:

```
SELECT *  
FROM machines;
```

Copied!

content_copy

You'll note that this query is not sufficient to perform the join and retrieve the information you need.

2. Complete the query to perform an inner join between the `machines` and `employees` tables on the `device_id` column. Replace `X` and `Y` with this column name:

```
SELECT *  
FROM machines  
INNER JOIN employees ON machines.X = employees.Y;
```

Copied!

content_copy

Note: *Placing the `employees` table after `INNER JOIN` makes it the right table.*

Note: If the output of the query is too wide for your shell, press the **Open Linux Console** button described in the Lab features section to open a full-screen view of the Bash shell, where you can re-enter the query.

How many rows did the inner join return?

185

132

124

85

Submit

Click **Check my progress** to verify that you have completed this task correctly.

Match employees to their machines

Check my progress

Task 2. Return more data

You now must return the information on all machines and the employees who have machines. Next, you must do the reverse and retrieve the information of all employees and any machines that are assigned to them.

To achieve this, you'll complete a left join and a right join on the employees and machines tables. The results will include all records from one or the other table. You must link these tables using the common `device_id` column.

1. Run the following SQL query to connect the machines and employees tables through a left join. You must replace the keyword X in the query:

```
SELECT *  
FROM machines  
X JOIN employees ON machines.device_id = employees.device_id;
```

Copied!

content_copy

Note: In a left join, all records from the table referenced after FROM and before LEFT JOIN are included in the result. In this case, all records from the machines table are included, regardless of whether they are assigned to an employee or not.

What is the value in the username column for the last record returned?

areyes

cgriffin

asundara

NULL

Submit

2. Run the following SQL query to connect the machines and employees tables through a right join. You must replace the keyword X in the query to solve the problem:

```
SELECT *  
FROM machines  
X JOIN employees ON machines.device_id = employees.device_id;
```

Copied!

content_copy

Note: In a right join, all records from the table referenced after RIGHT JOIN are included in the result. In this case, all records from the employees table are included, regardless of whether they have a machine or not.

What is the value in the username column for the last record returned?

asundara

NULL

areyes

cgriffin

Submit

Click **Check my progress** to verify that you have completed this task correctly.

Return more data

Check my progress

Task 3. Retrieve login attempt data

To continue investigating the security incident, you must retrieve the information on all employees who have made login attempts. To achieve this, you'll perform an inner join on the `employees` and `log_in_attempts` tables, linking them on the common `username` column.

- Run the following SQL query to perform an inner join on the `employees` and `log_in_attempts` tables. Replace X with the name of the right table. Then replace Y and Z with the name of the column that connects the two tables:

```
SELECT *  
FROM employees  
INNER JOIN X ON Y = Z;
```

Copied!

content_copy

Note: You must specify the table name with the column name (*table.column*) when joining the tables.

How many records are returned by this inner join?

210

200

145

175

Submit

Click **Check my progress** to verify that you have completed this task correctly.

Retrieve login attempt data

Check my progress

Conclusion

Great work!

You have completed this activity and should be able to use joins to combine data from multiple tables in a database.

You now have practical experience in using

- INNER JOIN,
- LEFT JOIN, and
- RIGHT JOIN.

Great work using SQL joins to obtain the precise data you need.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done.
Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments.
You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.

Exemplar: Complete a SQL join

1 hourFree

Activity overview

As a security analyst, you'll often find that you need data from more than one table.

Previously, you learned that a **relational database** is a structured database containing tables that are related to each other.

SQL joins enable you to combine tables that contain a shared column. This is helpful when you need to connect information that appears in different tables.

In this lab activity, you'll use SQL joins to connect separate tables and retrieve needed information.

Get ready to apply what you've learned and **join** some data!

Note: The terms **row** and **record** are used interchangeably.

Scenario

In this scenario, you'll investigate a recent security incident that compromised some machines.

You are responsible for getting the required information from the database for the investigation.

Here's how you'll do this task: **First**, you'll use an inner join to identify which employees are using which machines. **Second**, you'll use left and right joins to find machines that do not belong to any specific user and users who do not have any specific machine assigned to them. **Finally**, you'll use an inner join to list all login attempts made by all employees.

You're ready to join tables in SQL!

Note: In this lab you'll be working with the organization database and the tables it contains.

The lab starts with the organization database in the MariaDB shell that is already open.

This means you can start with the tasks as soon as you click the **Start Lab** button.

If you unintentionally exit the organization database in the MariaDB shell, you can reconnect by running the `sudo mysql organization` command.

Start your lab

Before you begin, you can review the instructions for using the Qwiklabs platform under the **Resources** tab in Coursera.

If you haven't already done so, click **Start Lab**. This brings up the terminal so that you can begin completing the tasks!

When you have completed all the tasks, refer to the **End your Lab** section that follows the tasks for information on how to end your lab.

Task 1. Match employees to their machines

First, you must identify which employees are using which machines. The data is located in the `machines` and `employees` tables.

You must use a SQL inner join to return the records you need based on a connecting column. In the scenario, both tables include the `device_id` column, which you'll use to perform the join.

1. Run the following query to retrieve all records from the `machines` table:

```
SELECT *  
FROM machines;
```

Copied!

content_copy

You'll note that this query is not sufficient to perform the join and retrieve the information you need.

2. Complete the query to perform an inner join between the machines and employees tables on the device_id column. Replace X and Y with this column name:

```
SELECT *  
FROM machines  
INNER JOIN employees ON machines.X = employees.Y;
```

Note: Placing the employees table after INNER JOIN makes it the right table.

To complete a join you need to link the joined tables on a common column. In the case of the employees and machines tables, the device_id column is common.

The correct query to solve this step:

```
SELECT *  
FROM machines  
INNER JOIN employees ON machines.device_id = employees.device_id;
```

Copied!

content_copy

Note: If the output of the query is too wide for your shell, press the **Open Linux Console** button described in the Lab features section to open a full-screen view of the Bash shell, where you can re-enter the query.

How many rows did the inner join return?

124

185

85

132

Submit

Answer: The inner join query returned 185 rows.

Click **Check my progress** to verify that you have completed this task correctly.

Match employees to their machines

Check my progress

Task 2. Return more data

You now must return the information on all machines and the employees who have machines. Next, you must do the reverse and retrieve the information of all employees and any machines that are assigned to them.

To achieve this, you'll complete a left join and a right join on the employees and machines tables. The results will include all records from one or the other table. You must link these tables using the common `device_id` column.

1. Run the following SQL query to connect the machines and employees tables through a left join. You must replace the keyword X in the query:

```
SELECT *
```

```
FROM machines
X JOIN employees ON machines.device_id = employees.device_id;
```

The correct query to solve this step:

```
SELECT *
FROM machines
LEFT JOIN employees ON machines.device_id = employees.device_id;
```

Copied!

content_copy

Note: In a left join, all records from the table referenced after FROM and before LEFT JOIN are included in the result. In this case, all records from the machines table are included, regardless of whether they are assigned to an employee or not.

What is the value in the username column for the last record returned?

NULL

cgriffin

asundara

areyes

Submit

Answer: The last username returned is NULL.

2. Run the following SQL query to connect the machines and employees tables through a right join. You must replace the keyword X in the query to solve the problem:

```
SELECT *
FROM machines
X JOIN employees ON machines.device_id = employees.device_id;
```

Note: In a right join, all records from the table referenced after RIGHT JOIN are included in the result. In this case, all records from the employees table are included, regardless of whether they have a machine or not.

The correct query to solve this step:

```
SELECT *  
FROM machines  
RIGHT JOIN employees ON machines.device_id = employees.device_id;
```

Copied!

content_copy

What is the value in the username column for the last record returned?

areyes

asundara

NULL

cgriffin

Submit

Answer: The value in the username column for the last record returned is areyes.

Click **Check my progress** to verify that you have completed this task correctly.

Return more data

Check my progress

Task 3. Retrieve login attempt data

To continue investigating the security incident, you must retrieve the information on all employees who have made login attempts. To achieve this, you'll perform an inner join

on the employees and log_in_attempts tables, linking them on the common username column.

- Run the following SQL query to perform an inner join on the employees and log_in_attempts tables. Replace X with the name of the right table. Then replace Y and Z with the name of the column that connects the two tables:

```
SELECT *  
FROM employees  
INNER JOIN X ON Y = Z;
```

Note: You must specify the table name with the column name (table.column) when joining the tables.

The correct query to solve this step:

```
SELECT *  
FROM employees  
INNER JOIN log_in_attempts ON employees.username =  
log_in_attempts.username;
```

Copied!

content_copy

How many records are returned by this inner join?

145

200

210

175

Submit

Answer: There are 200 records returned by the inner join.

Click **Check my progress** to verify that you have completed this task correctly.

Retrieve login attempt data

Check my progress

Conclusion

Great work!

You have completed this activity and should be able to use joins to combine data from multiple tables in a database.

You now have practical experience in using

- INNER JOIN,
- LEFT JOIN, and
- RIGHT JOIN.

Great work using SQL joins to obtain the precise data you need.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done.
Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments.
You can complete this if you choose to.
3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.