Exemplar: Perform an SQL query

Activity overview

Previously, you learned how to use basic SQL queries to retrieve information from a database. You have also learned about using the ORDER BY keyword to sort data returned in an ascending or a descending order.

In this lab activity, you'll use **SELECT** and **FROM** in SQL to return the information you need from a database. You'll also use the **ORDER BY** keyword to sequence the information returned by a query based on a specified column.

It's important to know how to query information from a database because this is a common task you might encounter as a security analyst. You should know how to get the information you need to improve security and keep data safe.

With that in mind, it's time to explore the scenario.

This exemplar is a walkthrough of the previous Qwiklab activity, including detailed instructions and solutions. You may use this exemplar if you were unable to complete the lab and/or you need extra guidance in competing lab tasks. You may also refer to this exemplar to prepare for the graded quiz in this module.

This exemplar is a walkthrough of the previous Qwiklab activity, including detailed instructions and solutions. You may use this exemplar if you were unable to complete the lab and/or you need extra guidance in competing lab tasks. You may also refer to this exemplar to prepare for the graded quiz in this module.

Note: *The terms* ***row*** *and* ***record*** *are used interchangeably in this lab activity.*

Scenario

In this scenario, you have to determine which employee devices must be updated. You also need to investigate user login activity to explore if any unusual activity has occurred.

The information you need is located in the **machines** and **login_attempts** tables in the **organization** database.

Here's how you'll do this task: **First**, you'll obtain information on the employee devices that must be updated. **Next**, you'll examine the login attempts for unusual activity. **Finally**, you'll use the **ORDER BY** keyword to sort the data returned by your SQL queries.

OK, let's get ready to practice running your very first SQL queries!

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.*

Task 1. Retrieve employee device data

In this task, you need to obtain information on employee devices because your team needs to update them. The information you need is in the **machines** table in the **organization** database.

First, you need to retrieve all the information about the employee devices.

1. Run the following query to select all device information from the **machines** table:

```
SELECT *
FROM machines;
```

Note: *Using the asterisk (**) returns all data from the specified table. Also, table names in MySQL are case-sensitive.*

The output returns all the contents of the **machines** table:

```
+-------------+-----------------+---------------+--------------+-------------+
| 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 |
|...          |                |               |              |           |
+-------------+-----------------+---------------+--------------+-------------+
```

200 rows in set (0.356 sec)

Next, you want to focus on the email client running on various devices.

2. Run the following query to select only the **device_id** and **email_client** columns from the machines table. Replace **X** with **device_id** and **Y** with **email_client**:

```
SELECT X, Y FROM machines;
```

The correct query to solve this step:

```
SELECT device_id, email_client
FROM machines;
```

The output should return only the selected columns of the machines table:

```
+--------------+----------------+
| device_id    | email_client   |
+--------------+----------------+
| a184b775c707 | Email Client 1 |
| a192b174c940 | Email Client 1 |
| a305b818c708 | Email Client 2 |
| a317b635c465 | Email Client 2 |
| a320b137c219 | Email Client 2 |
|...           |                |
+--------------+----------------+
200 rows in set (0.015 sec)
```

What email client is returned in the third row?

Answer: The email client returned in the third row is Email Client 2.

Now, you need information on the operating systems used on various devices and their last patch date.

3. Complete the query to return only the **device_id**, **operating_system**, and **OS_patch_date** columns from the **machines** table. Replace **X**, **Y**, and **Z** with the columns that you need to return:

```sql
SELECT X, Y, Z FROM machines;
```

The correct query to solve this step:

```sql
SELECT device_id, operating_system, OS_patch_date
FROM machines;
```

What is the patch date of the first entry?

Answer: The patch date of the first entry is 2021-09-01.

Task 2. Investigate login activity

In this task, you need to analyze the information from the **log_in_attempts** table to determine if any unusual activity has occurred.

First, you need to investigate the locations where login attempts were made to ensure that they're in expected areas (the United States, Canada, or Mexico).

1. Write a SQL query to select the **event_id** and **country** columns from the **log_in_attempts** table.

The correct query to solve this step:

```sql
SELECT event_id, country
FROM log_in_attempts;
```

Were any login attempts made from Australia?

Answer: No. Login attempts were not made from Australia.

Next, you need to check if login attempts were made outside of the organization's working hours.

2. Write a SQL query that selects the **username**, **login_date**, and **login_time** columns from the **log_in_attempts** table.

The correct query to solve this step:

```
SELECT username, login_date, login_time
FROM log_in_attempts;
```

What username is returned in the fifth row?

Answer: The username returned in the fifth row is jrafael.

Now, you need to get a complete picture of all login attempts.

3. Write a SQL query that selects all columns from the **log_in_attempts** table, using a single symbol after the **SELECT** keyword.

The correct query to solve this step:

```
FROM log_in_attempts;
```

Task 3. Order login attempts data

In this task, you need to use the **ORDER BY** keyword. You'll sequence the data that your query returns according to the login date and time.

First, you need to sort the information by date.

1. Run the following query, which orders **log_in_attempts** data by login_date**:**

```
FROM log_in_attempts
ORDER BY login_date;
SELECT *
```

What are the username and login date of the first record returned?

Answer: The first record returned contains a username of ivelasco and a login date of 2022-05-08.

Now, you need to further organize the previous results by ordering them by login_time.

2. Modify the query from the previous step by adding the login time to the **ORDER BY** clause. You must replace **X** with the appropriate column name:

```
1   ORDER BY login_date, X;
```

The correct query to solve this step:

```
1   ORDER BY login_date, login_time;
2   FROM log_in_attempts
3   SELECT *
```

What are the username and login date of the first record returned?

Answer: The first record returned contains a username of ivelasco and a login date of 2022-05-08.

Now, you need to further organize the previous results by ordering them by **login_time**.

2. Modify the query from the previous step by adding the login time to the **ORDER BY** clause. You must replace X with the appropriate column name:

```
1   SELECT *
2   FROM log_in_attempts
3   ORDER BY login_date, X;
```

The correct query to solve this step:

```sql
SELECT *
FROM log_in_attempts
ORDER BY login_date, login_time;
```

What are the username and login time of the first record returned by the above query?

Answer: The first record returned contains a username of bsand and a login time of 00:19:11.