Task 1. Retrieve login attempts after a certain date

In this task, you need to investigate a recent security incident. To do this, you need to gather information about login attempts made after a certain date.

1. Complete the SQL query to retrieve data for login attempts made after `'2022-05-09'`. Replace `X` with the correct operator:

```
SELECT *
FROM log_in_attempts
WHERE login_date X '2022-05-09';
```

SELECT * FROM log_in_attempts WHERE login_date > '2022-05-09';

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date > '2022-05-09';
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
|        6 | arutley  | 2022-05-12 | 17:00:59   | MEXICO  | 192.168.3.24    |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
|        9 | yappiah  | 2022-05-11 | 13:47:29   | MEX     | 192.168.59.136  |       1 |
|       10 | jrafael  | 2022-05-12 | 09:33:19   | CANADA  | 192.168.228.221 |       0 |
```

Now, based on your first query, you find a need to expand the date range to include 2022-05-09 in your search.

2. Complete the SQL query to retrieve data for login attempts that were made on or after `'2022-05-09'`. Replace `X` with the correct operator:

```
SELECT *
FROM log_in_attempts
WHERE login_date X '2022-05-09';
```

SELECT * FROM log_in_attempts WHERE login_date >= '2022-05-09';

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE 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 |
|        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 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
|        6 | arutley  | 2022-05-12 | 17:00:59   | MEXICO  | 192.168.3.24    |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
```

Task 2. Retrieve logins in a date range

In this task, you need to narrow the focus of the search. Login attempts made after 2022-05-11 shouldn't be included. Use the `BETWEEN` and `AND` operators to return results between `'2022-05-09'` and `'2022-05-11'`.

- Run the query to retrieve the required records. You must insert the required dates `X` and `Y`:

```
SELECT *
FROM log_in_attempts
WHERE login_date BETWEEN 'X' AND 'Y';
```

SELECT * FROM log_in_attempts WHERE login_date BETWEEN '2022-05-09' AND '2022-05-11';

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date BETWEEN '2022-05-09' AND '2022-05-11';
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
|        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 |
|        9 | yappiah  | 2022-05-11 | 13:47:29   | MEX     | 192.168.59.136  |       1 |
```

Task 3. Investigate logins at certain times

In this task, you need to investigate logins that were made at certain times. To do this, filter the data in the `log_in_attempts` table by login time (`login_time`).

First, your organization's typical work hours begin at 07:00:00. Retrieve all login attempts made before 07:00:00 to learn more about the users who are logging in outside of typical hours.

1. **Write a SQL query to retrieve data for login attempts made before `'07:00:00'`**

SELECT * FROM log_in_attempts WHERE login_time < '07:00:00';

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time < '07:00:00';
+----------+----------+------------+------------+---------+-----------------+----------+
| 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 |
|        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 |
```

The query in the previous step returned more results than required.

2. **Modify the query to return logins between `'06:00:00'` and `'07:00:00'`.**

SELECT * FROM log_in_attempts WHERE login_time BETWEEN '06:00:00' AND '07:00:00';

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time BETWEEN '06:00:00' AND '07:00:00';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
|       16 | mcouliba | 2022-05-11 | 06:44:22   | CAN     | 192.168.172.189 |       1 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       37 | eraab    | 2022-05-10 | 06:03:41   | CANADA  | 192.168.152.148 |       0 |
|       71 | mcouliba | 2022-05-09 | 06:57:42   | CAN     | 192.168.55.169  |       0 |
|       98 | gesparza | 2022-05-11 | 06:30:14   | CANADA  | 192.168.148.80  |       0 |
```

Task 4. Investigate logins by event ID

In this task, you need to investigate login attempts based on event ID numbers. With this query, you want to return only the `event_id`, `username`, and `login_date` fields from the `log_in_attempts` table.

Note: The `event_id` column contains numeric data; do not place numeric data in quotation marks.

1. **Write a query to return login attempts with `event_id` greater than or equal to `100`.**

SELECT event_id, username, login_date FROM log_in_attempts WHERE event_id >= 100;

```
MariaDB [organization]> SELECT event_id, username, login_date FROM log_in_attempts WHERE event_id >= 100;
+----------+----------+------------+
| event_id | username | login_date |
+----------+----------+------------+
|      100 | tmitchel | 2022-05-12 |
|      101 | sbaelish | 2022-05-08 |
|      102 | jreckley | 2022-05-09 |
|      103 | jhill    | 2022-05-11 |
|      104 | asundara | 2022-05-11 |
|      105 | cjackson | 2022-05-12 |
|      106 | tmitchel | 2022-05-12 |
```

The query in the previous step returned more data than required.

2. **Modify the query to return only login attempts with `event_id` between `100` and `150`.**

SELECT event_id, username, login_date FROM log_in_attempts WHERE event_id BETWEEN 100 AND 150;

```
MariaDB [organization]> SELECT event_id, username, login_date FROM log_in_attempts WHERE event_id BETWEEN 100 AND 150;
+----------+----------+------------+
| event_id | username | login_date |
+----------+----------+------------+
|      100 | tmitchel | 2022-05-12 |
|      101 | sbaelish | 2022-05-08 |
|      102 | jreckley | 2022-05-09 |
|      103 | jhill    | 2022-05-11 |
|      104 | asundara | 2022-05-11 |
|      105 | cjackson | 2022-05-12 |
|      106 | tmitchel | 2022-05-12 |
```

Conclusion

I have completed this activity and practiced applying

- **the `WHERE` keyword**
- **the `BETWEEN` and `AND` operators, and**
- **operators for working with numeric or date and time data types (for example, =, >, >=)**

to filter data from a table.