**Activity: Filter with AND, OR, and NOT**

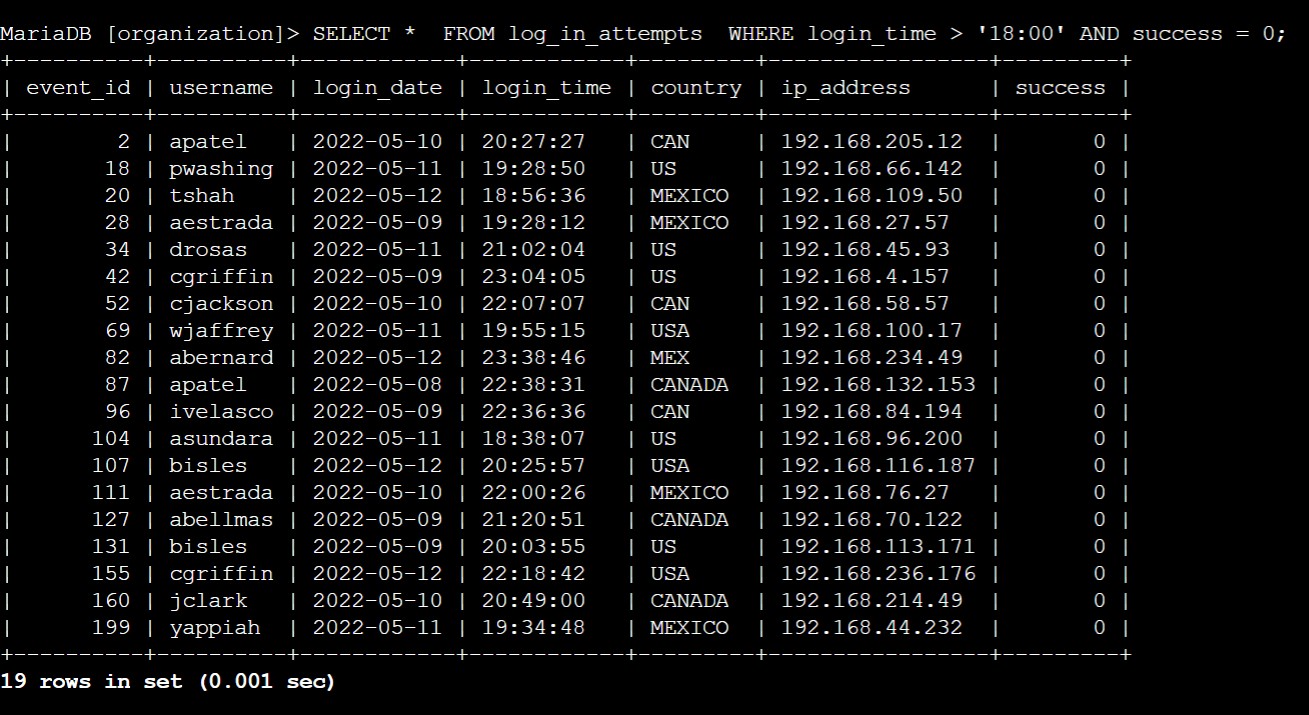
**Overview:** As a security analyst, you’ll likely need to analyse data. And often finding the specific data you’ll need depends on more than one factor. To retrieve specific pieces of information from the database, you can filter for multiple conditions. You can also filter for what does not match a particular condition. In this lab activity, you’ll use the AND, OR, and NOT operators to create more complex filters for SQL queries.

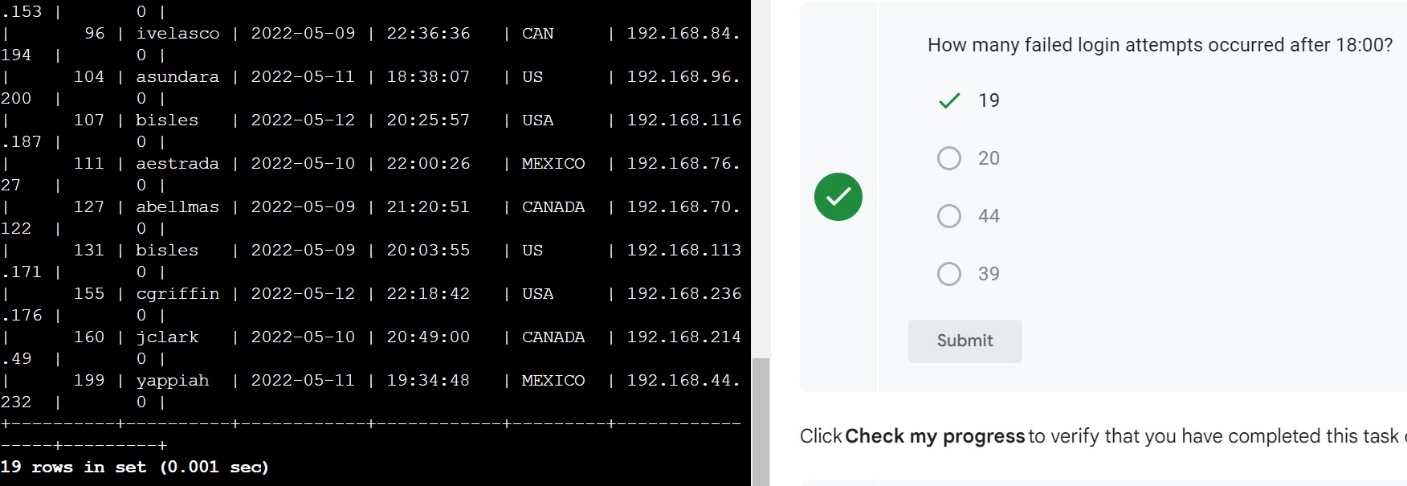
**Scenario:** In this scenario, you need to obtain specific information about employees, their machines, and the departments they belong to from the database. **First**, you’ll retrieve all failed login attempts after business hours. **Second**, you’ll retrieve all login attempts that occurred on specific dates. **Third**, you’ll retrieve logins that didn't originate in Mexico. **Fourth**, you’ll retrieve information about certain employees in the Marketing department. **Fifth**, you’ll retrieve information about employees in the Finance or the Sales department. **Finally**, you’ll obtain information about employees who are not in the Information Technology department.

**Start your lab:** click on “start lab” to start the lab.

**Task 1. Retrieve after hours failed login attempts**

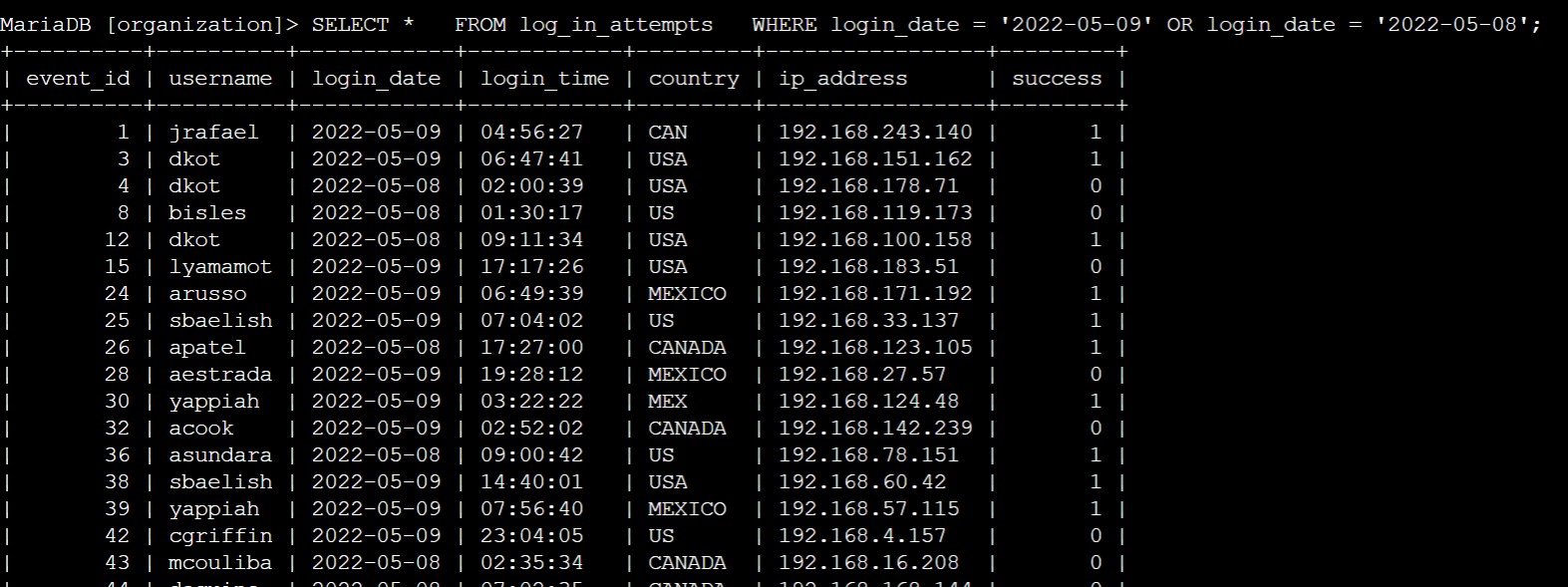
Your team is investigating failed login attempts that were made after business hours. You want to retrieve this information from the login activity. You’ll identify all unsuccessful attempts after 18:00. The login\_time column in the log\_in\_attempts table contains information on when login attempts were made. Office hours end at '18:00'. The success column in the log\_in\_attempts table contains values of TRUE or FALSE to indicate whether the login was successful. MySQL stores Boolean values as 1 for TRUE, and 0 for FALSE.

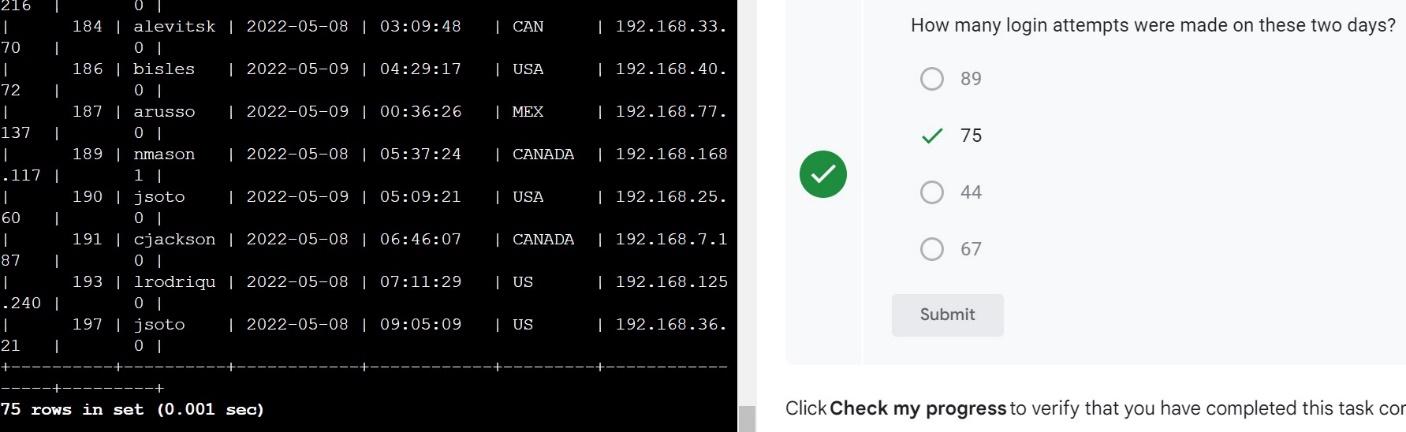
**$ SELECT \* FROM log\_in\_attempts WHERE login\_time > '18:00' AND success = 0 ;**



**Task 2. Retrieve login attempts on specific dates**

Your team is investigating a suspicious event that occurred on '2022-05-09'. You want to retrieve all login attempts that occurred on this day and the day before ('2022-05-08'). The login\_date column in the log\_in\_attempts table contains information on the dates when login attempts were made.

**$ SELECT \* FROM log\_in\_attempts WHERE login\_date = ' 2022-05-09' OR login\_date = ' 2022-05-08' ;**

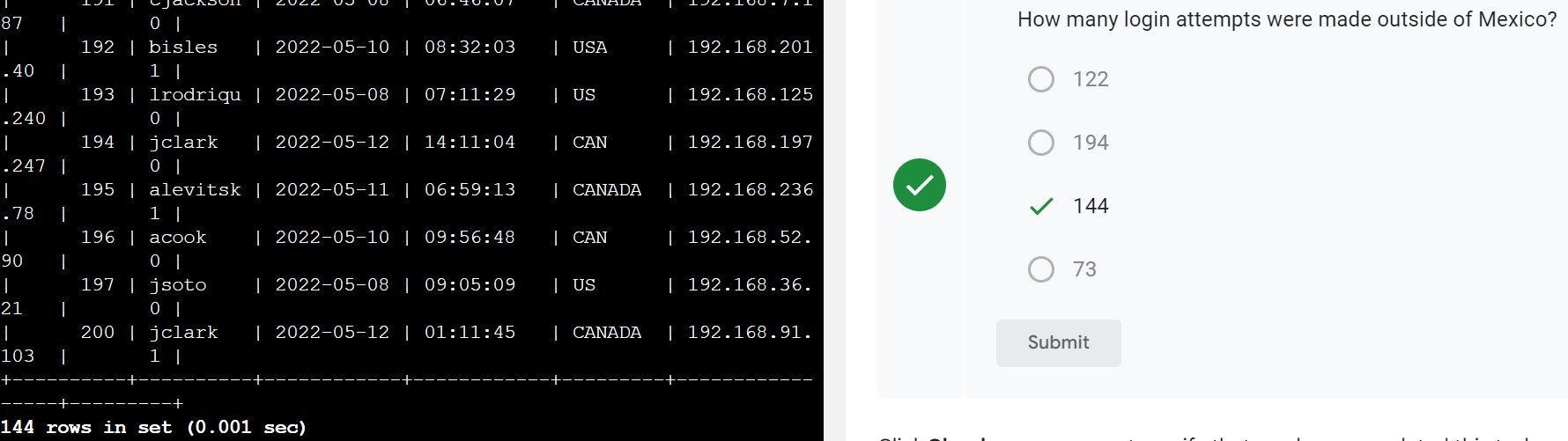


## **Task 3. Retrieve login attempts outside of Mexico**

Now, your team is investigating logins that did not originate in Mexico, and you need to find this information. Note that the country field includes entries with 'MEX' and 'MEXICO'. You should use the NOT and LIKE operators and the matching pattern 'MEX%'.

**$ SELECT \* FROM log\_in\_attempts WHERE NOT country LIKE 'MEX%' ;**

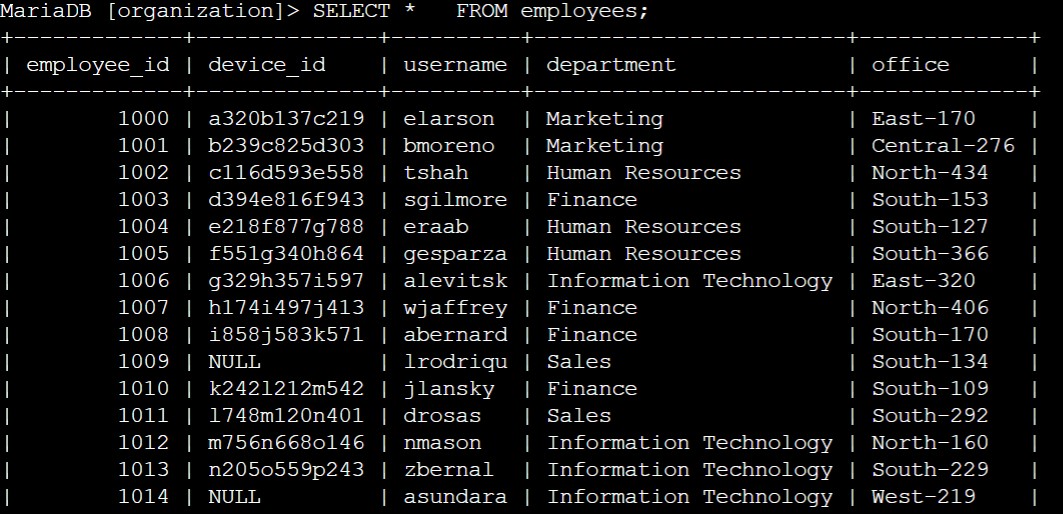




**Task 4. Retrieve employees in Marketing**

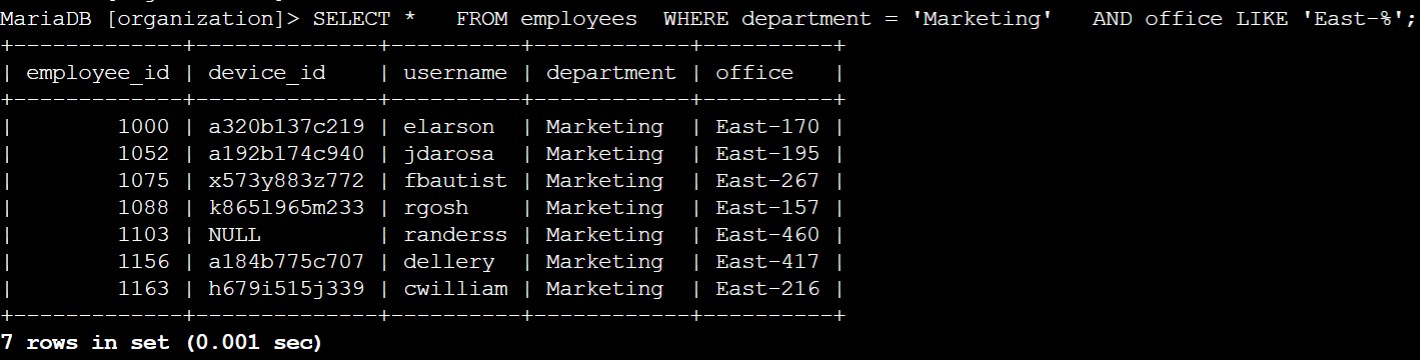
You can run the following SQL query if you need to view the columns and values in the employees table:

**$ SELECT \* FROM employees;**



Your team is updating employee machines, and you need to obtain the information about employees in the 'Marketing' department who are located in all offices in the East building (such as 'East-170' or 'East-320').

**$SELECT \* FROM employees WHERE department= ‘Marketing’ AND office LIKE ‘EAST-%’ ;**

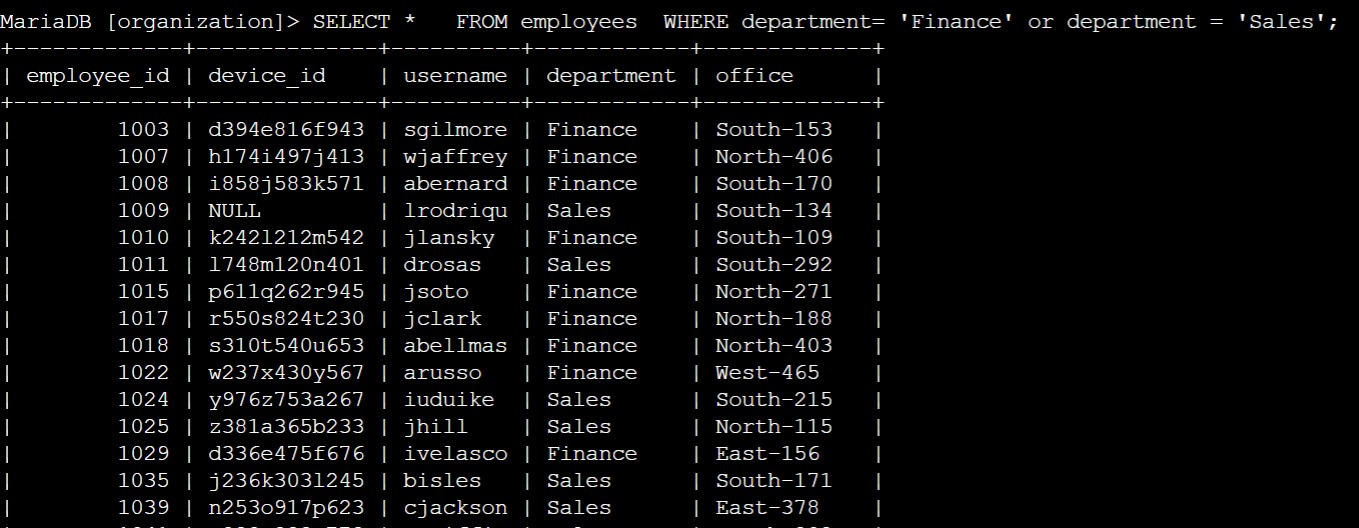


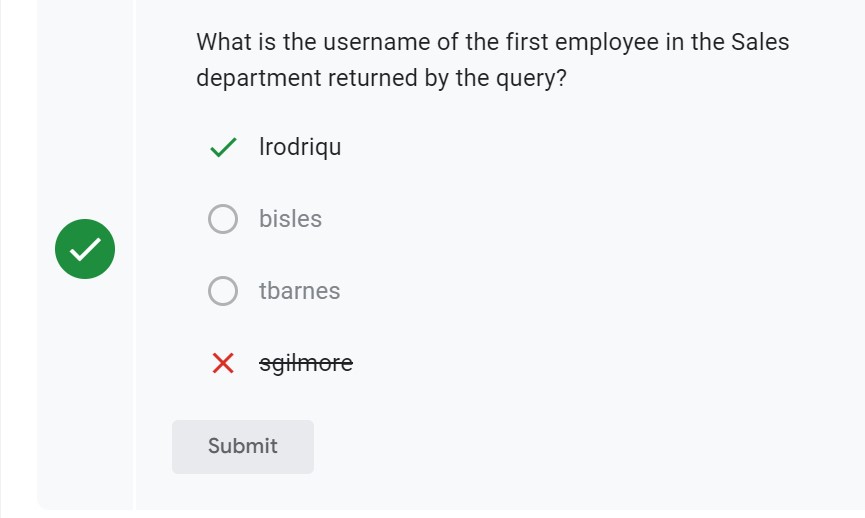
**Task 5. Retrieve employees in Finance or Sales**

Now, your team needs to perform a different update to the computers of all employees in the Finance or the Sales department, and you need to locate information on these employees.

Write a SQL query to retrieve records for employees in the 'Finance' or the 'Sales' department.

**$ SELECT \* FROM employees WHERE department=‘Finance’ or department=’Sales’ ;**



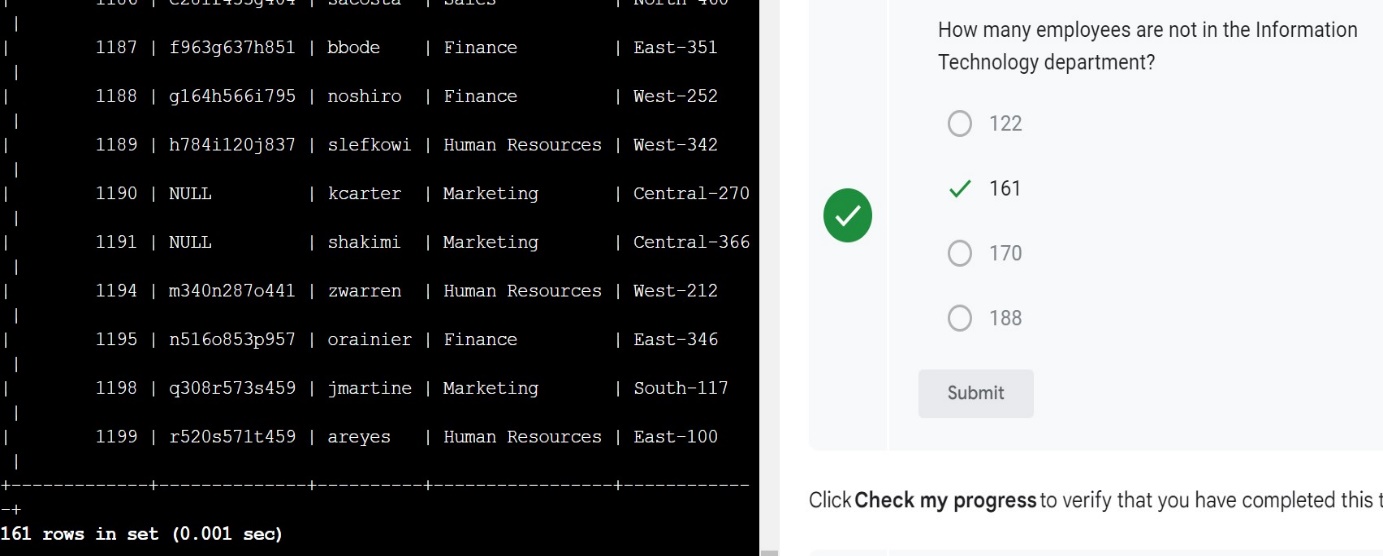


**Task 6. Retrieve all employees not in IT**

Your team needs to make one more update. This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department. You should use the NOT operator to identify these employees.

Write a SQL query to retrieve records for employees who are not in the 'Information Technology' department.

**$ SELECT \* FROM employees WHERE NOT department =‘Information Technology’ ;**



**Conclusion**

I now have practical experience in using SQL to

* run SQL queries to retrieve information from a database and
* apply AND, OR, and NOT operators to filter SQL queries.

