Apply filters to SQL queries

Project description

SQL (Structured Query Language) is a programming language that is used to communicate with and retrieve data from a database. There are several different flavours of SQL but most only differ in the syntax of the commands that are used to talk to the database and retrieve information.

This project provides details on a number of commands used to achieve different goals when retrieving data, and explanations relating to the commands used.

Retrieve after hours failed login attempts

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE
    -> login_time > '18:00'
    -> AND success = 0;
```

This query accesses the organisation database, and reads the table `log_in_attempts` to find the login attempts that were made after business hours. Business hours in this context means after 18:00 or 6:00 pm. From these login attempts, it returns only the attempts which were unsuccessful. To do this, we use the `>` operator to compare the `login_time` field to the desired value, and the `AND` operator to ensure that both the conditions that we specify must be true.

Retrieve login attempts on specific dates

```
MariaDB [organization]> SELECT * FROM log_in_attempts
    -> WHERE login_date = '2022-05-09'
    -> OR login_date = '2022-05-08';
```

This query accesses the organisation database and the table containing a record of all ogin attempts, and returns the attempts that were made on either the date `2022-5-09` or the date `2022-05-08`. The `OR` operator returns all the records where either one condition, or the other, or both have a value of true.

Retrieve login attempts outside of Mexico

```
MariaDB [organization]> SELECT * FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
```

This query requests all login attempts that occurred from a country that is not Mexico. Because “Mexico” can be represented in two different ways in the database, we use the `LIKE` operator in combination with a wildcard, written as a `%` in order to return only the entries where the country column starts with the string “`MEX`”. Then, because we want to return only the entries where the login attempt was from a country outside of Mexico, we negate this using the `NOT` operator.

Retrieve employees in Marketing

```
MariaDB [organization]> SELECT * FROM employees
    -> WHERE department = 'Marketing'
    -> AND office LIKE 'East%';
```

This query returns the information for all the employees that work in the Marketing department, but only if they also work in an office that is part of the East wing of the building. This is achieved by using the `WHERE` clause to specify the department as Marketing, and the `AND` operator is also used, along with the `LIKE` operator to return only the information for offices such as `East-101` or `East-325` that have been assigned to the Marketing department.

Retrieve employees in Finance or Sales

```
MariaDB [organization]> SELECT * FROM employees
    -> WHERE department = 'Finance'
    -> OR department = 'Sales';
```

This query returns the details of the employees that work in either the Finance department or in the Sales department. This is accomplished by using the `WHERE` clause to specify the department, as well as using the `OR` operator to return both the employees that work in the Finance department as well as the employees working in the Sales department.

Retrieve all employees not in IT

```
MariaDB [organization]> SELECT * FROM employees
    -> WHERE NOT department = 'Information Technology';
```

This query returns all the employees in the organisation that do not work in the Information Technology department. This is achieved by querying the database for all employees that work in the IT department, and then negating the selection using the `NOT` operator. The result is a list of all employees that work in every department, except for those who work in the Information Technology department.

Summary

This assignment focused on the execution of a number of database queries on a database containing information on employees working at a company. The information retrieved from the database could be used to inform decision-making about a specific department or perhaps about a specific employee within the company.