# Apply filters to SQL queries

## Project description

We have been requested to retrieve some data to help solve a security incident that happened after business hours. See the document *Table Formats* for the table descriptions. We shall use SQL queries to locate the data.

## Retrieve after hours failed login attempts

We need to locate failed login attempts that occurred after 18:00.

```
Select * from log_in_attempts where login_time>”18:00” and success=0;
```

## Retrieve login attempts on specific dates

We then attempt to retrieve login attempts on 2022-05-09 and the day before

```
Select * from log_in_attempts where login_date=”2022-05-08” OR
login_date=”2022-05-09”;
```

## Retrieve login attempts outside of Mexico

We have been informed that the attempt was not from Mexico, so we’ll filter that out.

```
Select * from log_in_attempts where NOT country like “MEX%”;
```

## Retrieve employees in Marketing

In a new task, we need to find all the employees in the Marketing department in the East building for an urgent update.

```
Select * from employees where department=”Marketing” and office like
“East%”;
```

## Retrieve employees in Finance or Sales

We now need to find all employees in Finance department or Sales department for a different update. .

```
Select * from employees where department=”Finance” or
department=”Sales”;
```

## Retrieve all employees not in IT

Finally, we need to find all employees not in IT department or “Information Technology”. They need the update.

```
Select * from employees where NOT department=”Information
Technology”;
```

## Summary

We have demonstrated proficient use of SQL filtering on strings, numbers, dates, and times. We have used NOT, AND, OR, and LIKE, to help us both filter log entries to narrow down suspects, and to make our update effort comprehensive and efficient.