# Project description

As a security professional at a large organization, I used SQL to investigate potential security issues involving login attempts and employee activity. I retrieved specific records from log\_in\_attempts and employees tables using SQL filters. These queries helped detect failed login attempts after hours, analyze suspicious dates, and review employees by department and office location to inform security updates.

# Retrieve after hours failed login attempts

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM log\_in\_attempts  
WHERE success = 0 AND login\_time > '18:00:00';  
```  
  
\*\*Explanation\*\*  
This query retrieves all rows from the log\_in\_attempts table where login attempts failed (success = 0) and occurred after 18:00 (6 PM). This allows us to identify suspicious failed login activity happening outside regular business hours.

# Retrieve login attempts on specific dates

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM log\_in\_attempts  
WHERE login\_date = '2022-05-08' OR login\_date = '2022-05-09';  
```  
  
\*\*Explanation\*\*  
This query filters the log\_in\_attempts table to return any login attempts that happened on May 8th or May 9th of 2022. It’s useful for investigating login behavior around a known suspicious incident date.

# Retrieve login attempts outside of Mexico

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM log\_in\_attempts  
WHERE country NOT LIKE '%MEX%';  
```

\*\*Explanation\*\*  
The NOT LIKE '%MEX%' condition filters out any rows where the country column contains either MEX or MEXICO, ensuring we only retrieve login attempts from outside Mexico. The % wildcard allows for partial string matching.

# Retrieve employees in Marketing

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM employees  
WHERE department = 'Marketing' AND office LIKE 'East%';  
```  
  
\*\*Explanation\*\*  
This query retrieves all employees in the Marketing department who are located in offices that start with "East" (e.g., East-170, East-320). The LIKE 'East%' pattern is used to match any office in the East building.

# Retrieve employees in Finance or Sales

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM employees  
WHERE department = 'Finance' OR department = 'Sales';  
```  
  
\*\*Explanation\*\*  
This query retrieves all employees who are in either the Finance or Sales departments. The OR operator ensures that any row matching either condition is returned.

# Retrieve all employees not in IT

\*\*SQL Query\*\*  
```sql  
SELECT \* FROM employees  
WHERE department != 'Information Technology';  
```  
  
\*\*Explanation\*\*  
This query returns all employees whose department is not 'Information Technology'. The != operator is used to exclude a specific value, identifying who still needs a security update that IT has already received.

# Summary

In this project, I used SQL to filter and retrieve data relevant to security investigations. I identified after-hours failed login attempts, isolated suspicious dates, excluded known safe locations, and gathered employee records based on department and office location. Through the use of WHERE, AND, OR, NOT, and LIKE filters, I successfully created targeted queries to assist with improving system security and operational awareness.