#3 SQL Injection - SQLi

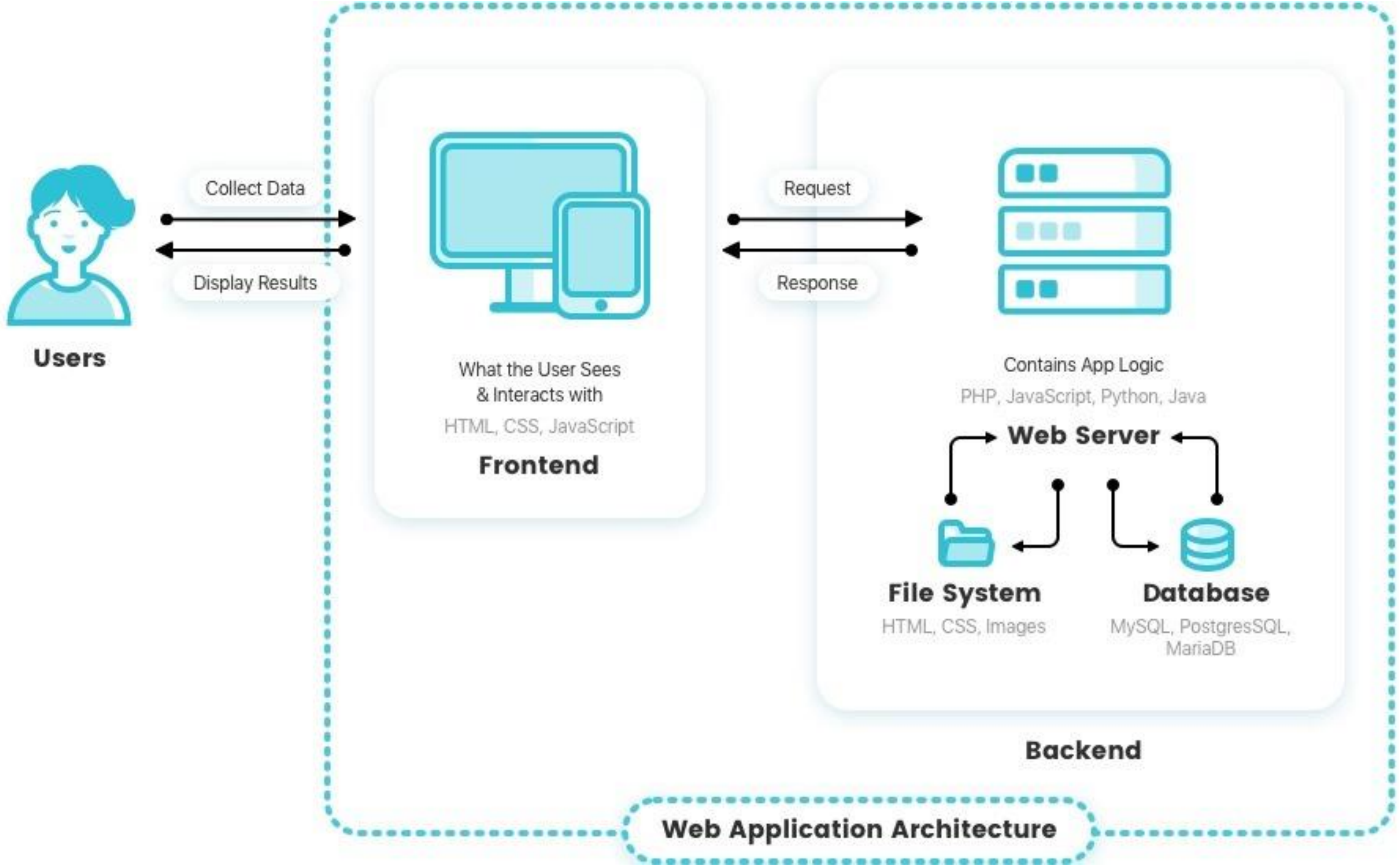
SEGURANÇA NAS ORGANIZAÇÕES E INFORMÁTICA

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

2024/2025

Operation Ecosystem

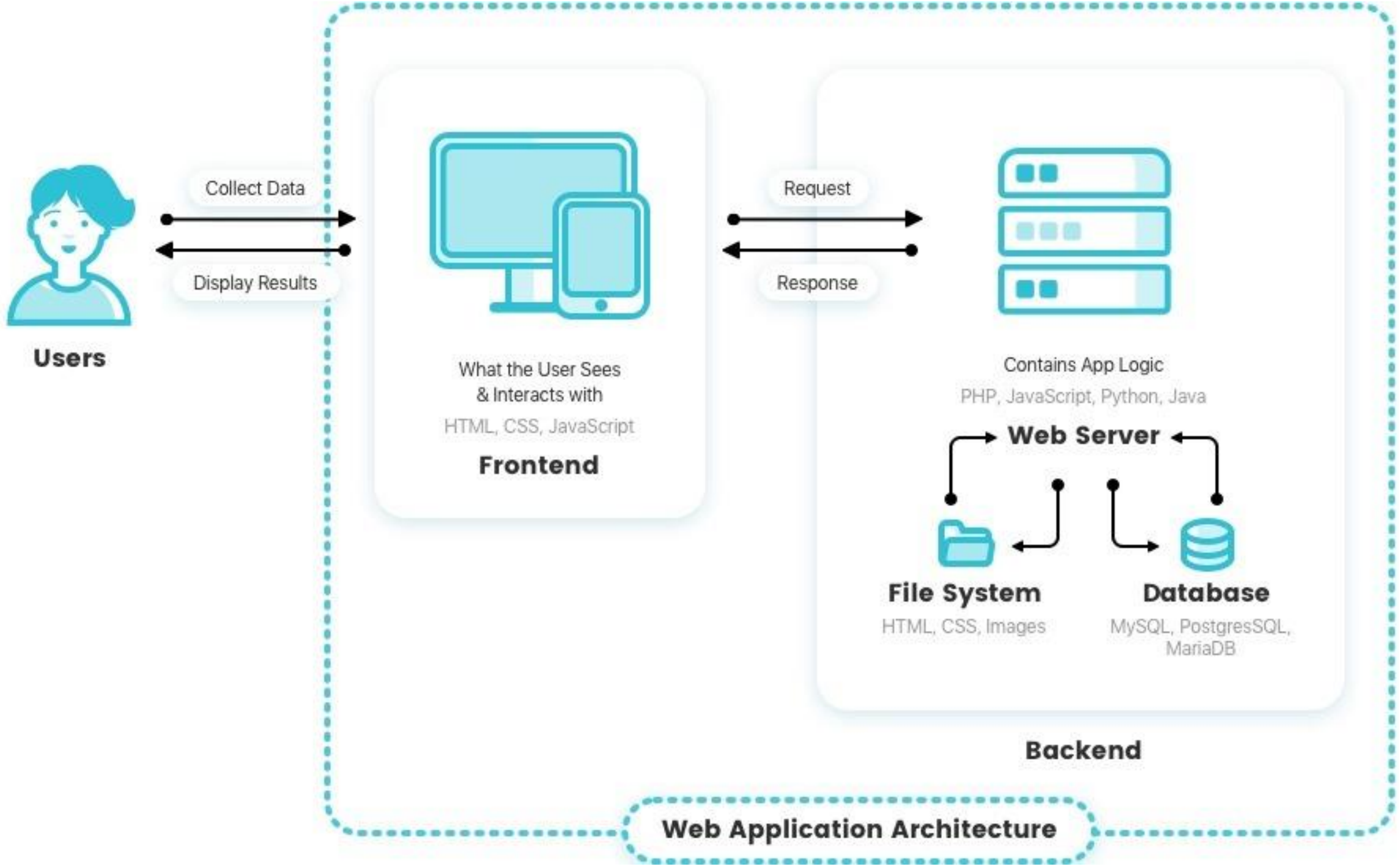
Web Architecture



Reference: https://reinvently.com/blog/fundamentals-web-application-architecture/

Operation Ecosystem

Web Architecture

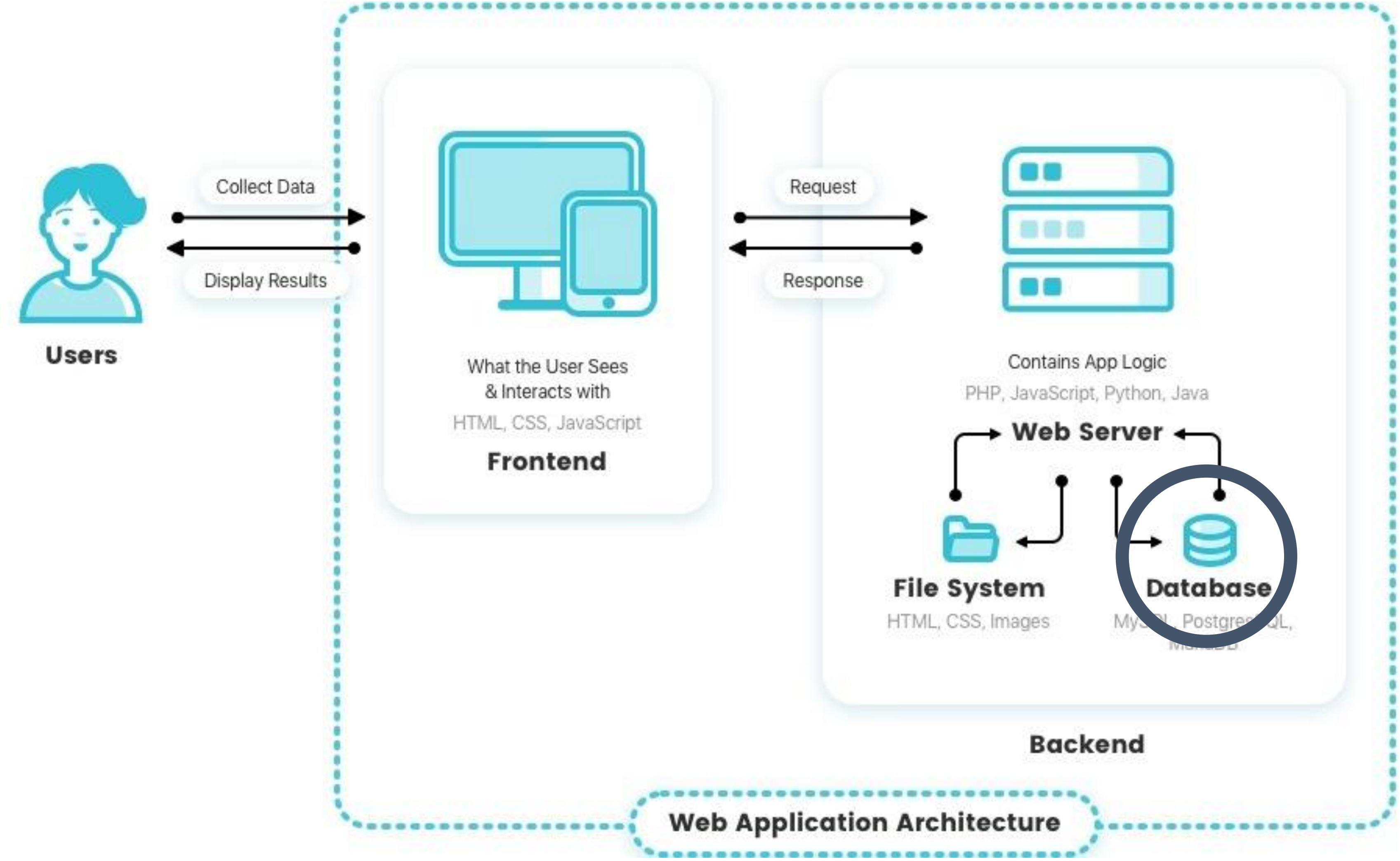


Each subsystem is prone to vulnerabilities, which can lead to attacks

Reference: https://reinvently.com/blog/fundamentals-web-application-architecture/

Operation Ecosystem

Web Architecture



SQL Injections are one type of web vulnerabilities which targets SQL databases

Reference: https://reinvently.com/blog/fundamentals-web-application-architecture/

SQL Injection

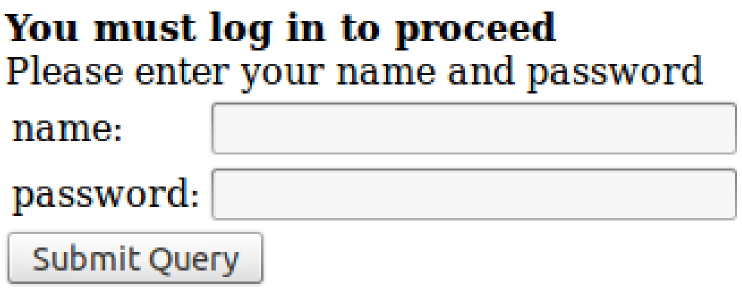
Why it happens

The vulnerability is usually present because of lack of or wrongly done sanity checks

Loading code or text that shouldn’t be loaded, or should be checked

SQL Injection

Example



You must log in to proceed

Please enter your name and password

name:

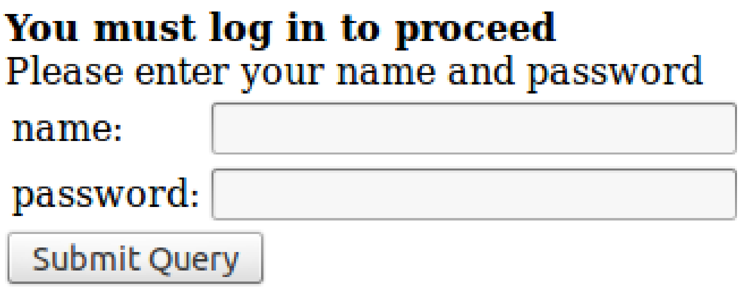
password:

The developer created a SQL query to verify if the username and password exists in the database

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

SQL Injection

Example



You must log in to proceed

Please enter your name and password

name:

password:

The developer created a SQL query to verify if the username and password exists in the database

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

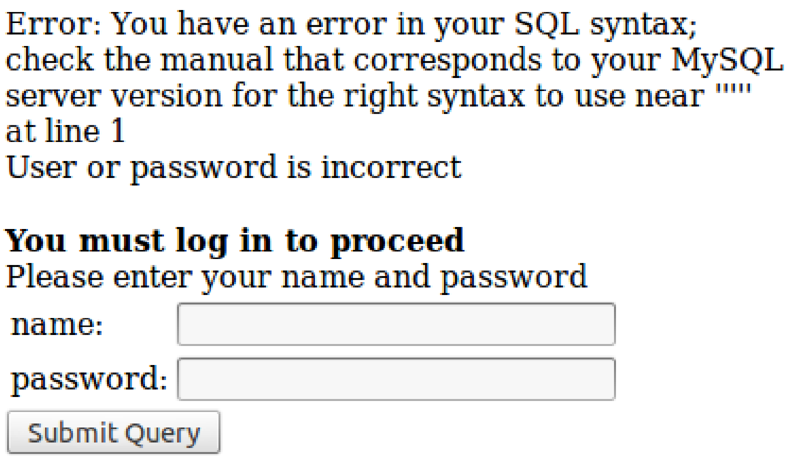
SQL Injection

Example

What if my name is ‘ ?

```
user = "'"
```

```
SELECT user FROM users WHERE user=''' AND password='<any value>'
```



Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''''' at line 1
User or password is incorrect

You must log in to proceed
Please enter your name and password

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

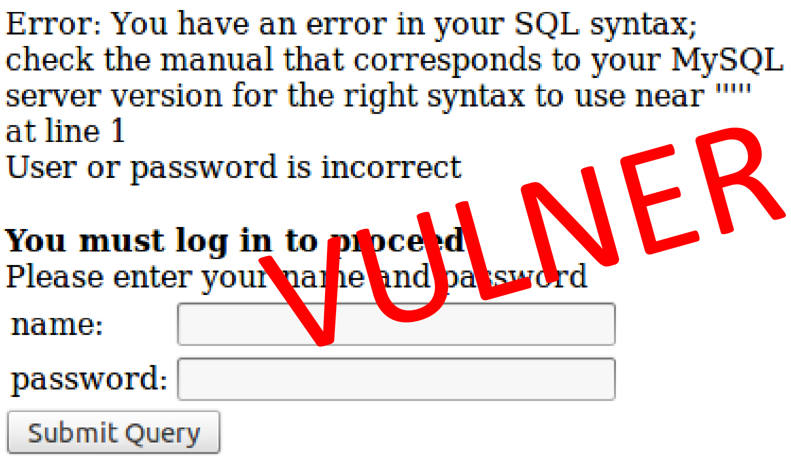
SQL Injection

Example

What if my name is ‘ ?

```
user = "'"
```

```
SELECT user FROM users WHERE user=''' AND password='<any value>'
```



Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "''" at line 1
User or password is incorrect

You must log in to proceed
Please enter your name and password

name:

password:

VULNERABLE!

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

SQL Injection

A more significant attack

Let’s play with the user name:

```
user = "admin'; -- //"
```

```
SELECT user FROM users WHERE user='admin'; -- //' AND password='<any value>'
```

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

SQL Injection

A more aggressive attack

Let’s play with the user name:

```
user = "'; DROP TABLE users; -- //"
```

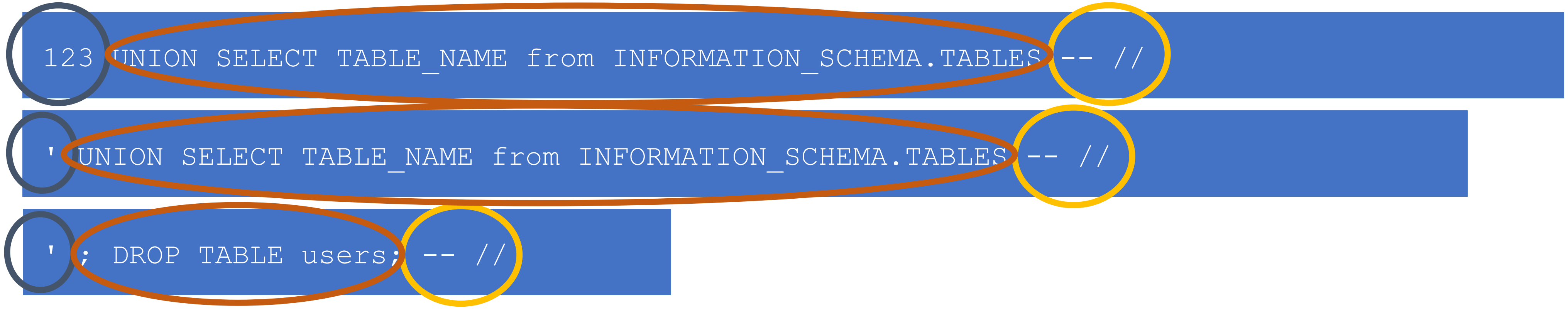
```
SELECT user FROM users WHERE user=''; DROP TABLE users; -- //' AND password='<any value>'
```

```
query = "SELECT user FROM users WHERE user='" + user + "' AND password='" + password + "'"
```

SQL Injection

Typical operation model

- Escape from current context
  - If context is a string, close string
  - If context is a number, insert a number
- Insert the new query component and/or a new query
- End with the the comment characters to avoid errors



```
123 UNION SELECT TABLE_NAME from INFORMATION_SCHEMA.TABLES -- //
' UNION SELECT TABLE_NAME from INFORMATION_SCHEMA.TABLES -- //
' ; DROP TABLE users; -- //
```

SQL Injection Mitigations

Strategies to prevent SQLi

- **Validate all input parameters**
  - Username can only have letters
  - Emails must comply with RFC 2822
  - Avoid block listing validation!
  - OWASP Input validation Cheat Sheet [1]
- **Use prepared statements**
  - The SQL query is pre-compiled and, therefore, is independent from the input data
- **Review the OWASP SQL Injection Prevention Cheat Sheet** [2]
  - Loads of information
  - Code review and developer guides

[1] https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html
[2] https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html