Exemplo de Código Python vulnerável SQl Injection

```
import sqlite3

username = input("Username: ")
password = input("Password: ")

conn = sqlite3.connect('database.db')
cursor = conn.cursor()

query = f"SELECT * FROM users WHERE username = '{username}' AND password = '{password}'"
cursor.execute(query)

if cursor.fetchone():
    print("Login successful!")
else:
    print("Invalid credentials.")
```

Neste exemplo, se o usuário inserir **' OR '1'='1** como username e password, a consulta SQL será:

```
SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '' OR '1'='1'
```

Isso fará com que o login sempre seja bem-sucedido, independentemente das credenciais inseridas. Para prevenir essa vulnerabilidade, é recomendado usar consultas parametrizadas ou frameworks que lidam com a sanitização de dados automaticamente.

Exemplo seguro usando consultas parametrizadas:

```
import sqlite3

username = input("Username: ")
password = input("Password: ")

conn = sqlite3.connect('database.db')
cursor = conn.cursor()

query = "SELECT * FROM users WHERE username = ? AND password = ?"
cursor.execute(query, (username, password))

if cursor.fetchone():
    print("Login successful!")
else:
    print("Invalid credentials.")
```