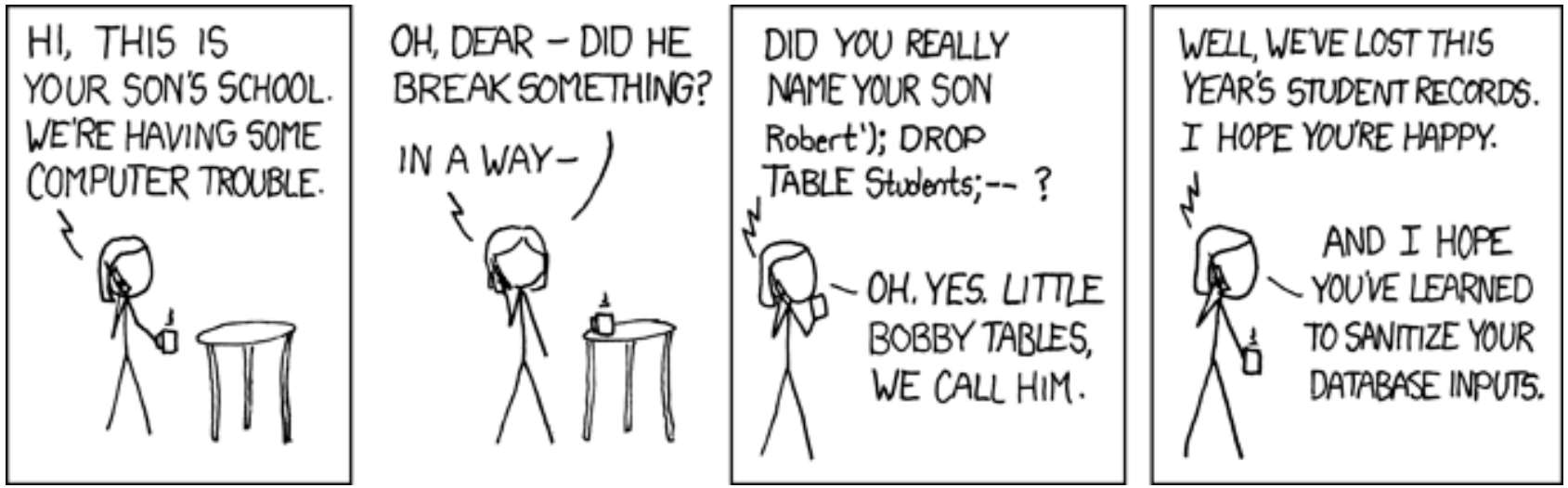
Exploits of a Mom...



https://www.xkcd.com/327/

Demo!

Travis can log in as Alice via SQL injection...

(Advanced) Computer Security!

Network & Web Security

SQL Injection Attacks & Countermeasures

(part I)

Prof. Travis Peters

Montana State University

CS 476/594 – Computer Security

Spring 2021

https://www.traviswpeters.com/cs476

Today

Reminder!
Please update your Slack, GitHub, Zoom
(first/last name, professional photo/background)

- Announcements
  - Lab 03 due today
  - Lab 04 released

- Learning Objectives
  - Basics of the Internet -> towards web apps
  - A (brief) SQL tutorial
  - The SQL Injection (SQLi) attack with examples
  - Understanding SQLi vulnerabilities
  - Understanding SQLi countermeasures

Warning:
Never target websites that you do not have explicit permission to "test" against.

All of our work targets webapps deployed in containers that run within your SEED Labs VM

Internet/Web Crash Course!

Activity: Reflect/Review!

How does the Internet work? How does the Web work? Important Concepts?

- "Handshakes" between clients/servers
- 

- IP addresses / DNS servs
- Protocols (OSI model)
  TCP/IP model
  — HTTP = GET, POST

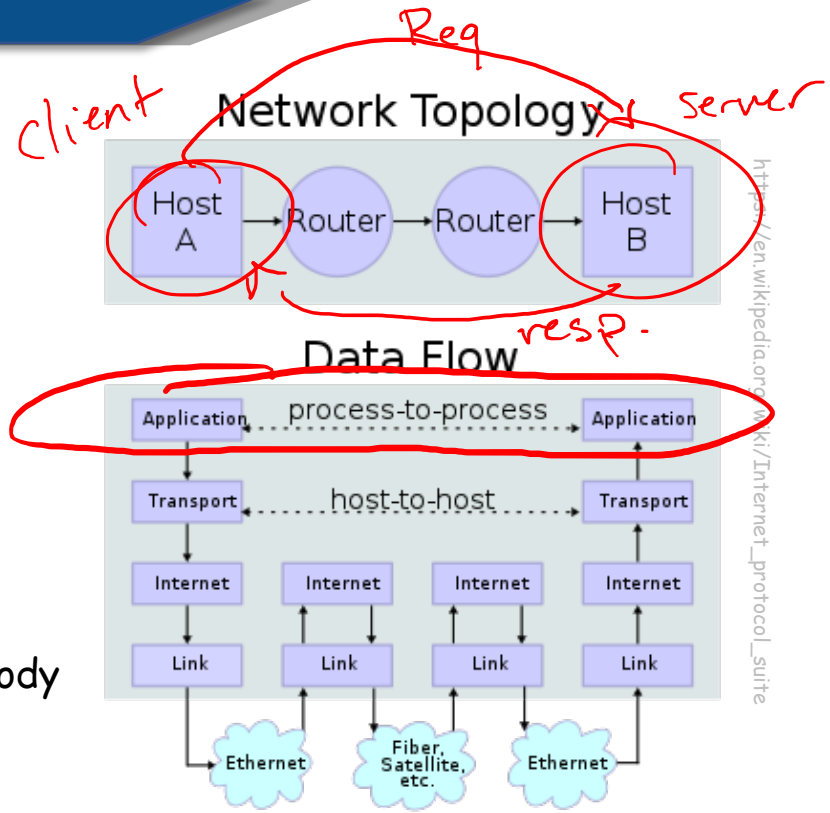
[header(s)]
[data/body]

Activity: Reflect/Review!

Watch:

How Does the Internet Work? (~8min.)
https://www.youtube.com/watch?v=x3c1ih2NJEg

Activity: Reflect/Review!

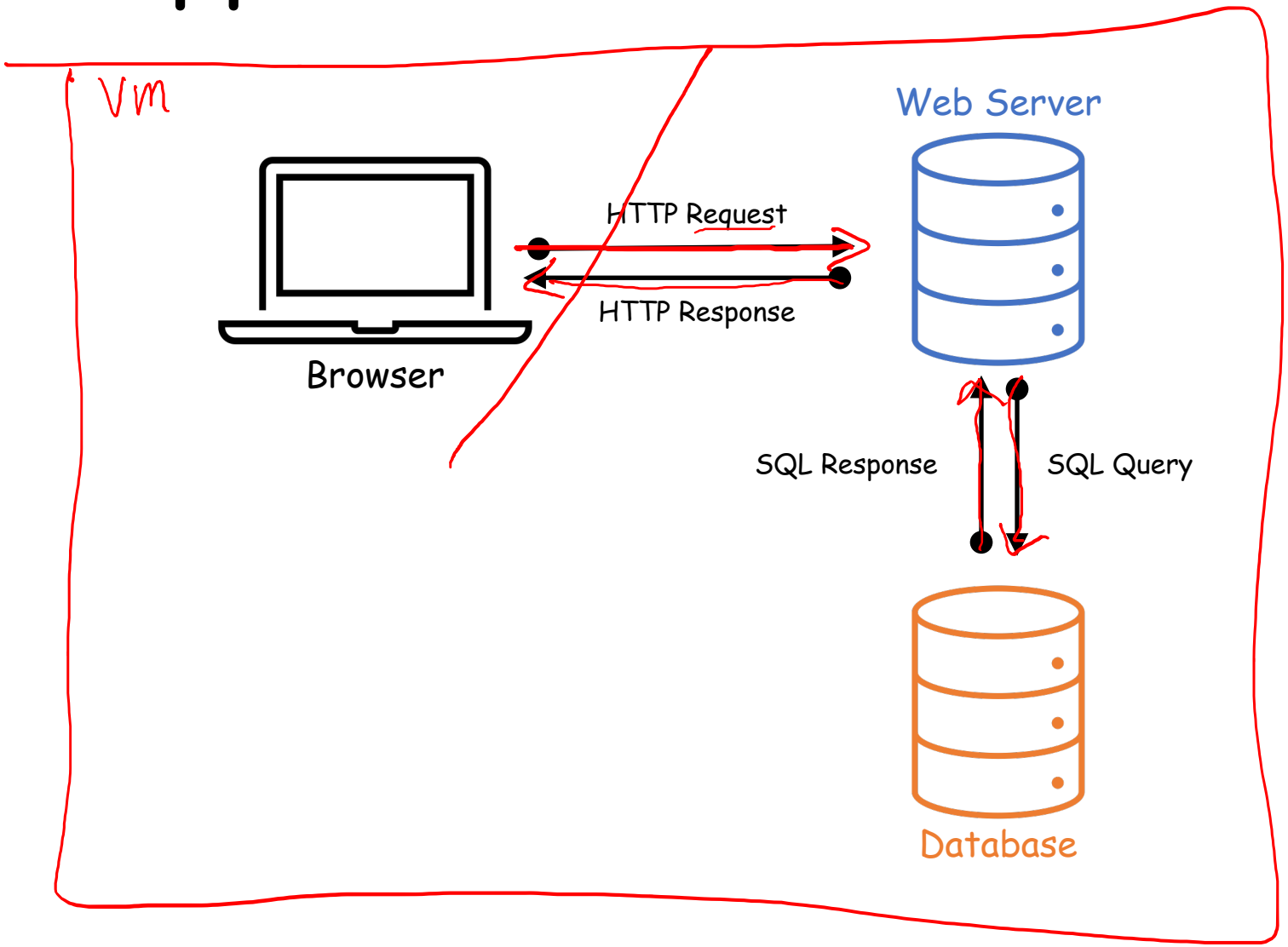


- How does the Internet work? How does the Web work?
- Communicating on the web
  - TCP/IP, HTTP
  - URLs
    - protocol://hostname[:port]/[path/]file[?param1=value1&param2=value2...]
  - HTTP Requests
    - **Format:** Request Method / params, Request Headers, [blank line], Request Body
    - **Methods:** GET, POST, HEAD, UPDATE, etc.
    - **Headers:** Host, Referer, User-Agent, Cookie, ...
  - HTTP Responses
    - **Format:** Status Line, Response Headers, [blank line], Response Body
    - **Status Codes:** 1XX (info), 2XX (successful request), 3XX (redirect), 4XX (bad request), 5XX (server error)
    - **Headers:** Server, Set-Cookie, Expires, Content-Type, Content-Length
- Server-side functionality
  - Serve Static Resources: HTML, CSS, images, client-side Javascript, etc.
  - Serve Dynamic Resources: (^) + PHP, Python, Ruby, Java, Javascript (NodeJS), shellscripts
  - Datastores (Databases):
    - Relational (e.g., MySQL, Postgres) vs.
    - Non-Relational (e.g., MongoDB, DocumentDB)

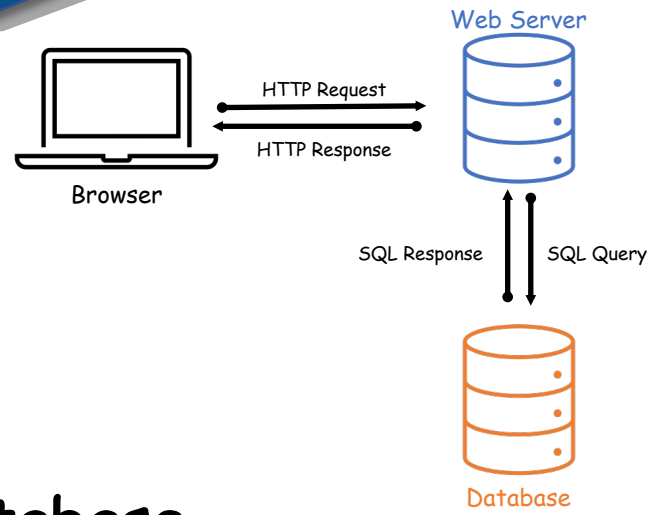
Web Apps

A high-level look + our lab setup

Web Apps!


VM
Web Server
HTTP Request
HTTP Response
Browser
SQL Response
SQL Query
Database

Our Setup



- See Lab 04 for details
  - Update/create your local copy of the class code
  - Use Docker Compose to bring up the **webapp + mysql database**
  - **Website:**
    - Visit http://www.seedlabsqlinjection.com (within the VM!)
    - Log in as one of the pre-setup SEED Labs users

10.0.9.5

/etc/hosts

- Next:
  - Get familiar with HTTP Header Live (Simple Firefox extension)
  - [Or.... Install Burp Suite Community Edition (Legit Web Sec Tool!)]

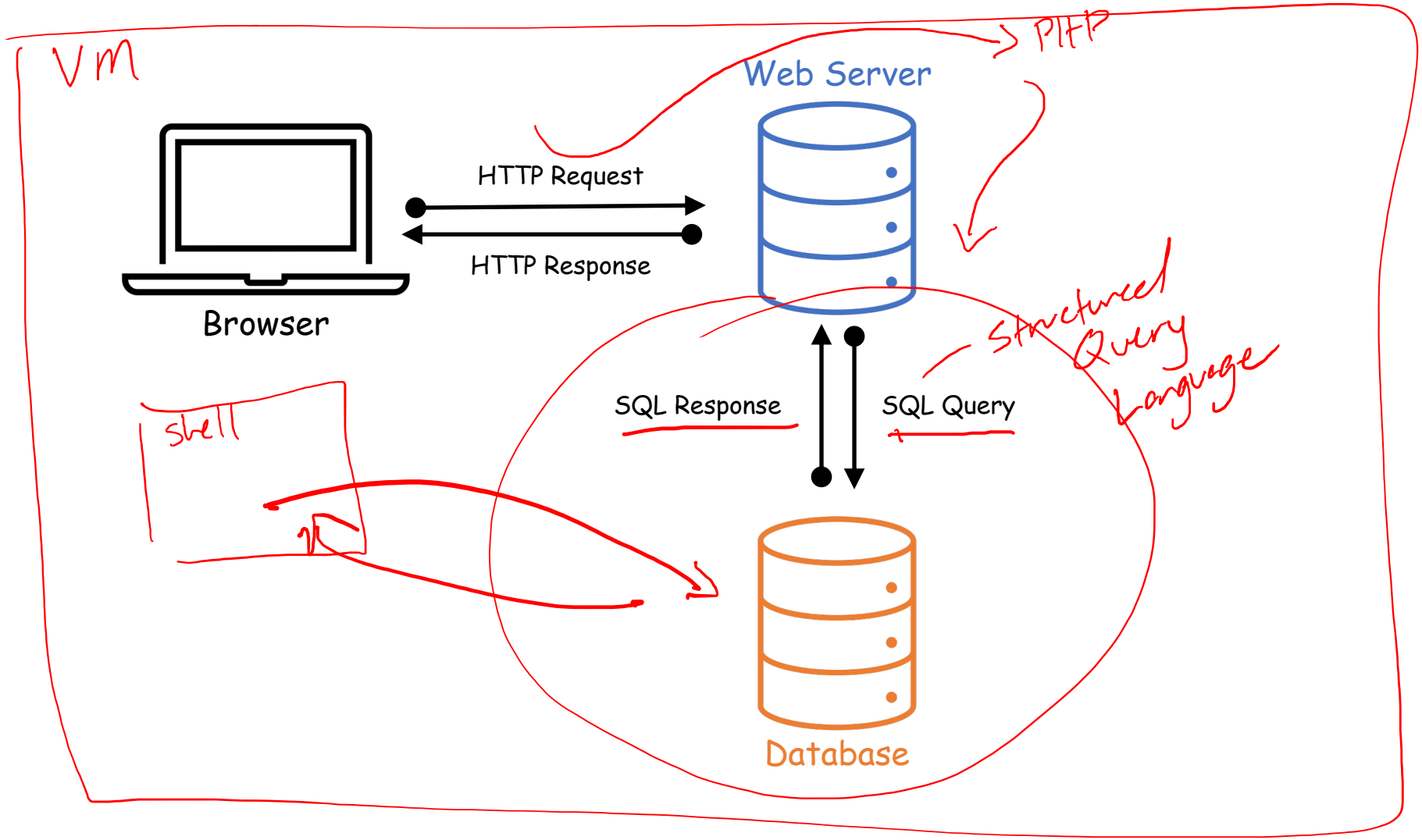
CONTENT ADVISORY:

Don't visit the lab websites outside of the VMs.
Unfortunately, sites with explicit content have been setup.
I nor the authors of SEED Labs condone these sites.

SQL Tutorial

A whirlwind tour of some basic (but important!) SQL concepts

SQL Tutorial



```
$ mysql --user=root --password=dees
Welcome to the MySQL monitor.
mysql>
```

```
mysql> show databases;
...
mysql> CREATE DATABASE dbtest;
Query OK, 1 row affected (0.00 sec)
```

Create a database

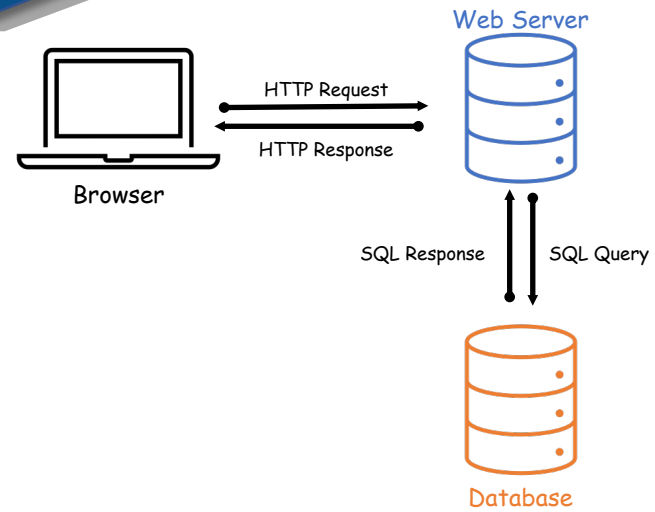
```
mysql> USE dbtest
Database changed
mysql> CREATE TABLE employee (
    -> ID       INT (6) NOT NULL AUTO_INCREMENT,
    -> Name     VARCHAR (30) NOT NULL,
    -> EID      VARCHAR (7) NOT NULL,
    -> Password VARCHAR (60),
    -> Salary   INT (10),
    -> SSN      VARCHAR (11),
    -> PRIMARY KEY (ID)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

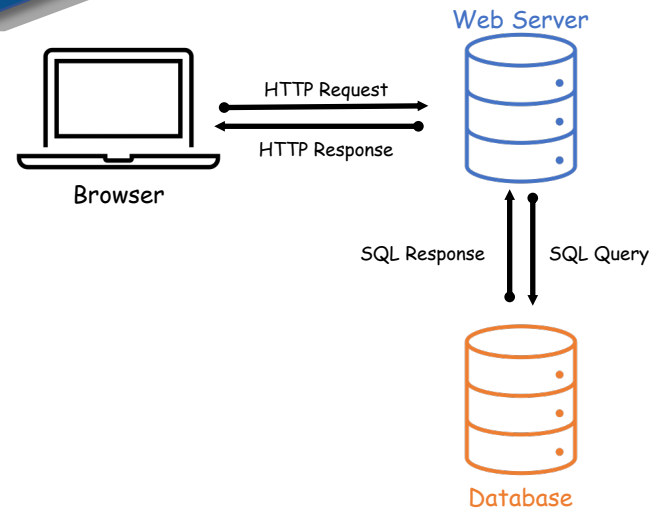
Indicate which database to use

Create a table

```
mysql> DESCRIBE employee;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| ID       | int(6)      | NO   | PRI | NULL    | auto_increment |
| Name     | varchar(30) | NO   |     | NULL    |                |
| EID      | varchar(7)  | NO   |     | NULL    |                |
| Password | varchar(60) | YES  |     | NULL    |                |
| Salary   | int(10)     | YES  |     | NULL    |                |
| SSN      | varchar(11) | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)
```

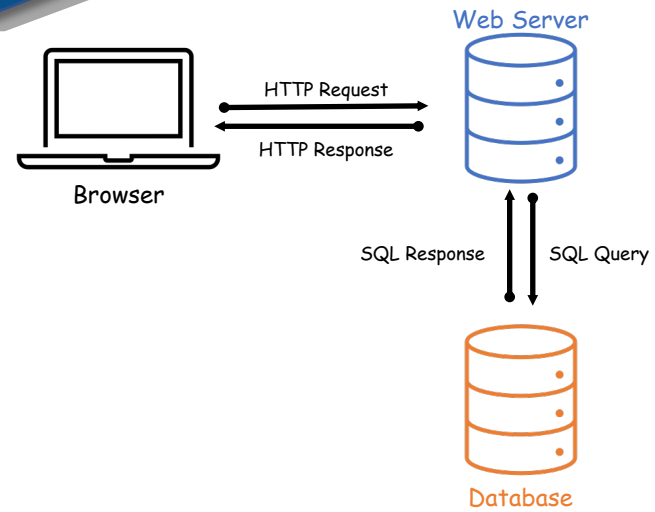
Examine the structure of a table





Inserting a new record into a table

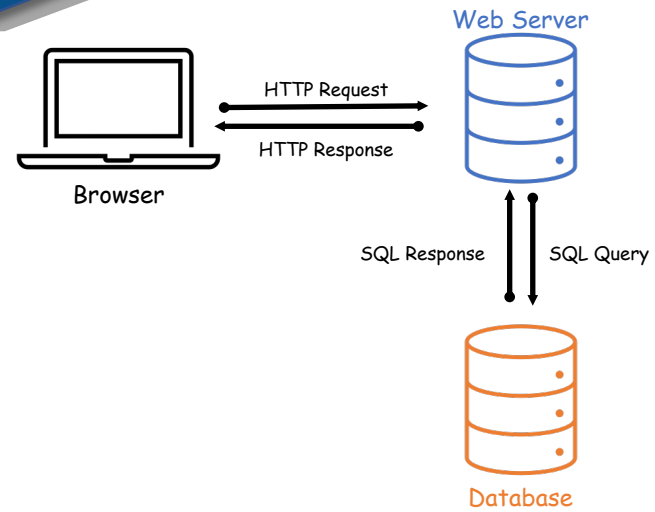
```
mysql> INSERT INTO employee (Name, EID, Password, Salary, SSN)
       VALUES ('Ryan Smith', 'EID5000', 'paswd123', 80000, '555-55-5555');
Query OK, 1 row affected (0.00 sec)


#...repeat a few times to add more entries...
INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Alice', 'EID5000', 'paswd123', 80000, '555-55-5555');
INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Bob', 'EID5001', 'paswd123', 80000, '555-66-5555');
INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Charlie', 'EID5002', 'paswd123', 80000, '555-77-5555');
INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('David', 'EID5003', 'paswd123', 80000, '555-88-5555');
```



everything!

Selecting entries from a table

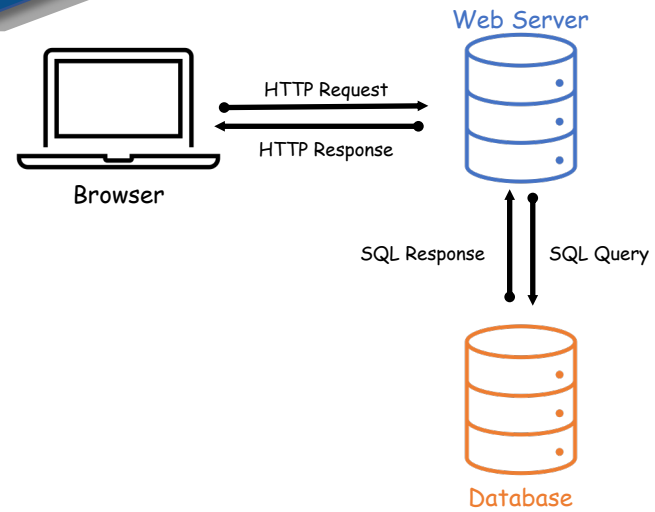
```
mysql> SELECT * FROM employee;
+----+------------+---------+----------+--------+-------------+
| ID | Name       | EID     | Password | Salary | SSN         |
+----+------------+---------+----------+--------+-------------+
|  1 | Ryan Smith | EID5000 | paswd123 |  80000 | 555-55-5555 |
|  2 | Alice      | EID5000 | paswd123 |  80000 | 555-55-5555 |
|  3 | Bob        | EID5001 | paswd123 |  80000 | 555-66-5555 |
|  4 | Charlie    | EID5002 | paswd123 |  80000 | 555-77-5555 |
|  5 | David      | EID5003 | paswd123 |  80000 | 555-88-5555 |
+----+------------+---------+----------+--------+-------------+
5 rows in set (0.00 sec)
```



Selecting entries from a table conditionally with the WHERE clause

```
mysql> SELECT * FROM employee WHERE EID='EID5001';
+----+------+---------+----------+--------+-------------+
| ID | Name | EID     | Password | Salary | SSN         |
+----+------+---------+----------+--------+-------------+
|  3 | Bob  | EID5001 | paswd123 |  80000 | 555-66-5555 |
+----+------+---------+----------+--------+-------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM employee WHERE EID='EID5001' OR Name='David';
+----+-------+---------+----------+--------+-------------+
| ID | Name  | EID     | Password | Salary | SSN         |
+----+-------+---------+----------+--------+-------------+
|  3 | Bob   | EID5001 | paswd123 |  80000 | 555-66-5555 |
|  5 | David | EID5003 | paswd123 |  80000 | 555-88-5555 |
+----+-------+---------+----------+--------+-------------+
2 rows in set (0.00 sec)
```



Selecting entries from a table conditionally with the WHERE clause

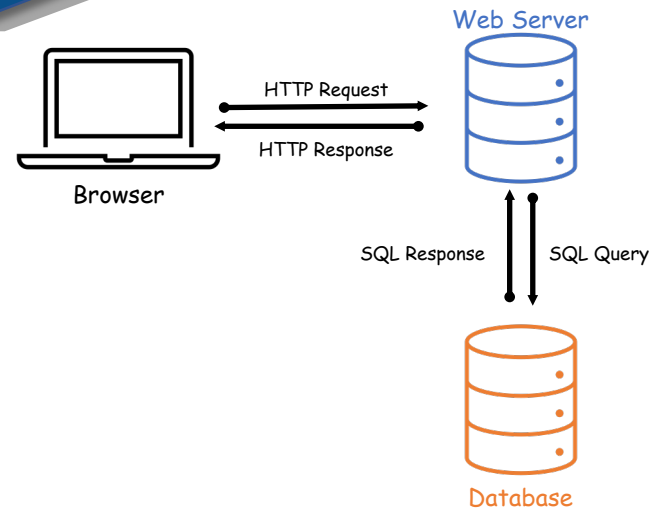
```
mysql> SELECT * FROM employee WHERE EID='EID5001';
+----+------+---------+----------+--------+-------------+
| ID | Name | EID     | Password | Salary | SSN         |
+----+------+---------+----------+--------+-------------+
|  3 | Bob  | EID5001 | paswd123 |  80000 | 555-66-5555 |
+----+------+---------+----------+--------+-------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM employee WHERE EID='EID5001' OR Name='David';
+----+-------+---------+----------+--------+-------------+
| ID | Name  | EID     | Password | Salary | SSN         |
+----+-------+---------+----------+--------+-------------+
|  3 | Bob   | EID5001 | paswd123 |  80000 | 555-66-5555 |
|  5 | David | EID5003 | paswd123 |  80000 | 555-88-5555 |
+----+-------+---------+----------+--------+-------------+
2 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM employee WHERE 1=1;
```

ALWAYS TRUE

Q: What happens when we run this statement?

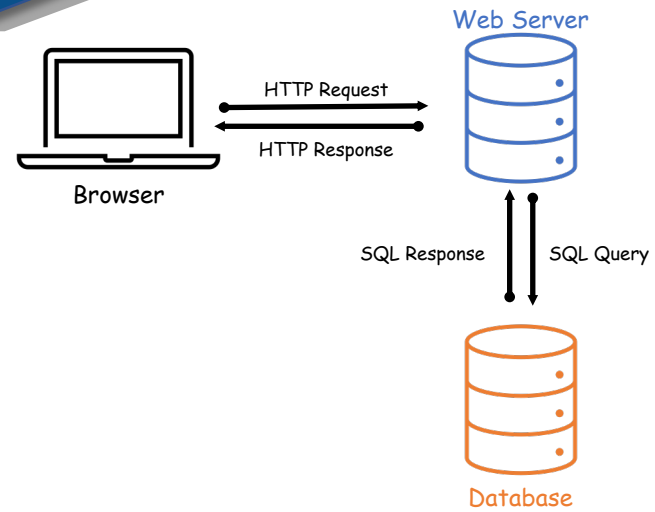


Selecting entries from a table conditionally with the WHERE clause

```
mysql> SELECT * FROM employee WHERE EID='EID5001';
+----+------+---------+----------+--------+-------------+
| ID | Name | EID     | Password | Salary | SSN         |
+----+------+---------+----------+--------+-------------+
|  3 | Bob  | EID5001 | paswd123 |  80000 | 555-66-5555 |
+----+------+---------+----------+--------+-------------+
1 row in set (0.00 sec)
```

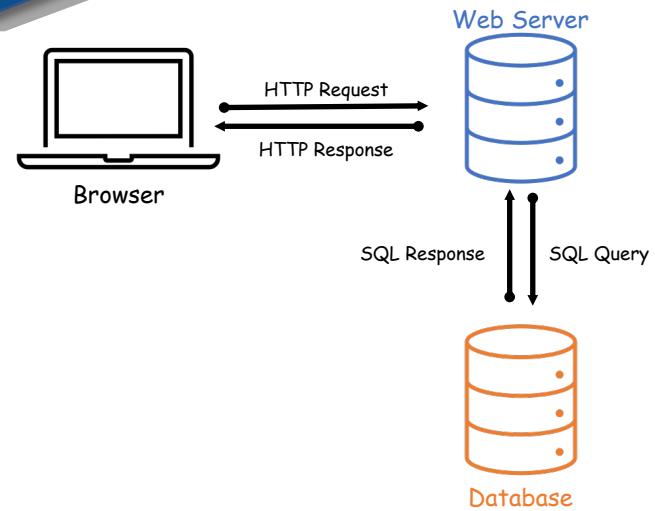
```
mysql> SELECT * FROM employee WHERE EID='EID5001' OR Name='David';
+----+-------+---------+----------+--------+-------------+
| ID | Name  | EID     | Password | Salary | SSN         |
+----+-------+---------+----------+--------+-------------+
|  3 | Bob   | EID5001 | paswd123 |  80000 | 555-66-5555 |
|  5 | David | EID5003 | paswd123 |  80000 | 555-88-5555 |
+----+-------+---------+----------+--------+-------------+
2 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM employee WHERE 1=1;
+----+------------+---------+----------+--------+-------------+
| ID | Name       | EID     | Password | Salary | SSN         |
+----+------------+---------+----------+--------+-------------+
|  1 | Ryan Smith | EID5000 | paswd123 |  80000 | 555-55-5555 |
|  2 | Alice      | EID5000 | paswd123 |  80000 | 555-55-5555 |
|  3 | Bob        | EID5001 | paswd123 |  80000 | 555-66-5555 |
|  4 | Charlie    | EID5002 | paswd123 |  80000 | 555-77-5555 |
|  5 | David      | EID5003 | paswd123 |  80000 | 555-88-5555 |
+----+------------+---------+----------+--------+-------------+
5 rows in set (0.00 sec)
```



Updating entries in a table

```
mysql> UPDATE employee SET SALARY=82000 WHERE Name='Bob';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM employee WHERE Name='Bob';
+----+------+---------+----------+--------+-------------+
| ID | Name | EID     | Password | Salary | SSN         |
+----+------+---------+----------+--------+-------------+
|  3 | Bob  | EID5001 | paswd123 |  82000 | 555-66-5555 |
+----+------+---------+----------+--------+-------------+
1 row in set (0.00 sec)
```



Comments in SQL

```
mysql> SELECT * FROM employee;  # Comment to the end of line
mysql> SELECT * FROM employee;  -- Comment to the end of line
mysql> SELECT * FROM /* inline comment */ employee;
```

But Travis, why have a whole slide dedicated to talking about comments?!?!?!?!

Oh you'll see...

You Try!

```
$ mysql --user=root --password=dees
mysql> show databases;
mysql> CREATE DATABASE dbtest;
mysql> USE dbtest
mysql> CREATE TABLE employee (
    -> ID       INT (6) NOT NULL AUTO_INCREMENT,
    -> Name     VARCHAR (30) NOT NULL,
    -> EID      VARCHAR (7) NOT NULL,
    -> Password VARCHAR (60),
    -> Salary   INT (10),
    -> SSN      VARCHAR (11),
    -> PRIMARY KEY (ID)
    -> );
mysql> DESCRIBE employee;

# INSERT Statements
mysql> INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Alice', 'EID5000', 'paswd123', 80000, '555-55-5555');
mysql> INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Bob', 'EID5001', 'paswd123', 80000, '555-66-5555');
mysql> INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('Charlie', 'EID5002', 'paswd123', 80000, '555-77-5555');
mysql> INSERT INTO employee (Name, EID, Password, Salary, SSN) VALUES ('David', 'EID5003', 'paswd123', 80000, '555-88-5555');

# SELECT Statements
mysql> SELECT * FROM employee;
mysql> SELECT * FROM mytest WHERE Name='Bob';
mysql> SELECT * FROM employee WHERE EID='EID5001' OR Name='David';
mysql> SELECT * FROM employee WHERE 1=1;

# Experiment with comments! "#" "--" "/* ... */"
```