MPCS 53001: Databases

Zach Freeman

Lecture #8

Convo Between Two Cool People

SNL has a new cast member named Luke Null. How does their database handle this? I guess that's why they do "IS NULL"

Imagining his first day:

“Hi, I’m Luke.”

"Last name?"

"Null“

"Okay: SELECT * FROM Performer WHERE last_name IS NULL.... sorry Luke, looks like you're not in our database.“

"No no no - my last name IS Null"

"Yep that's what I searched. You're not there."

Yup there you go... That was so super nerdy.

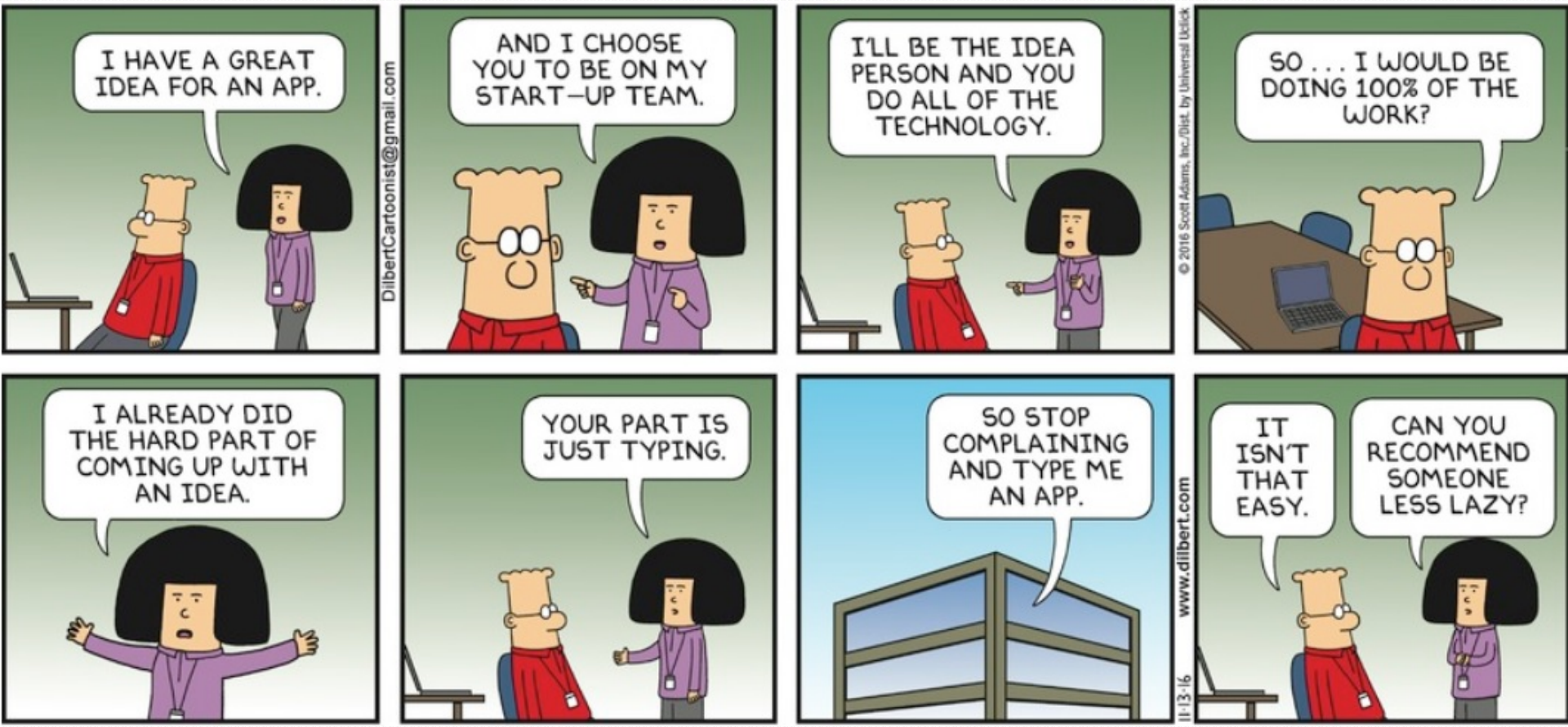
Reminder

- Next class:
  - THIS Saturday (11/18) and NEXT Tuesday (11/21)
  - Data warehousing (w/guest speaker Frank Greco from IBM)

Homework

- Assignment 8 (project and Gradiance lab)
  - Get web account set up.
  - Basic web programming (PHP/HTML with MySQL interactions) – retrieving/displaying data.
  - Assistance with PHP and web programming will be given during office hours.

Great App Idea



Overview For Today

- Web system architecture
- Database connectivity
- PHP
- SQL Injection

Web System Architecture

- Three-tier (or multi-tier) architecture

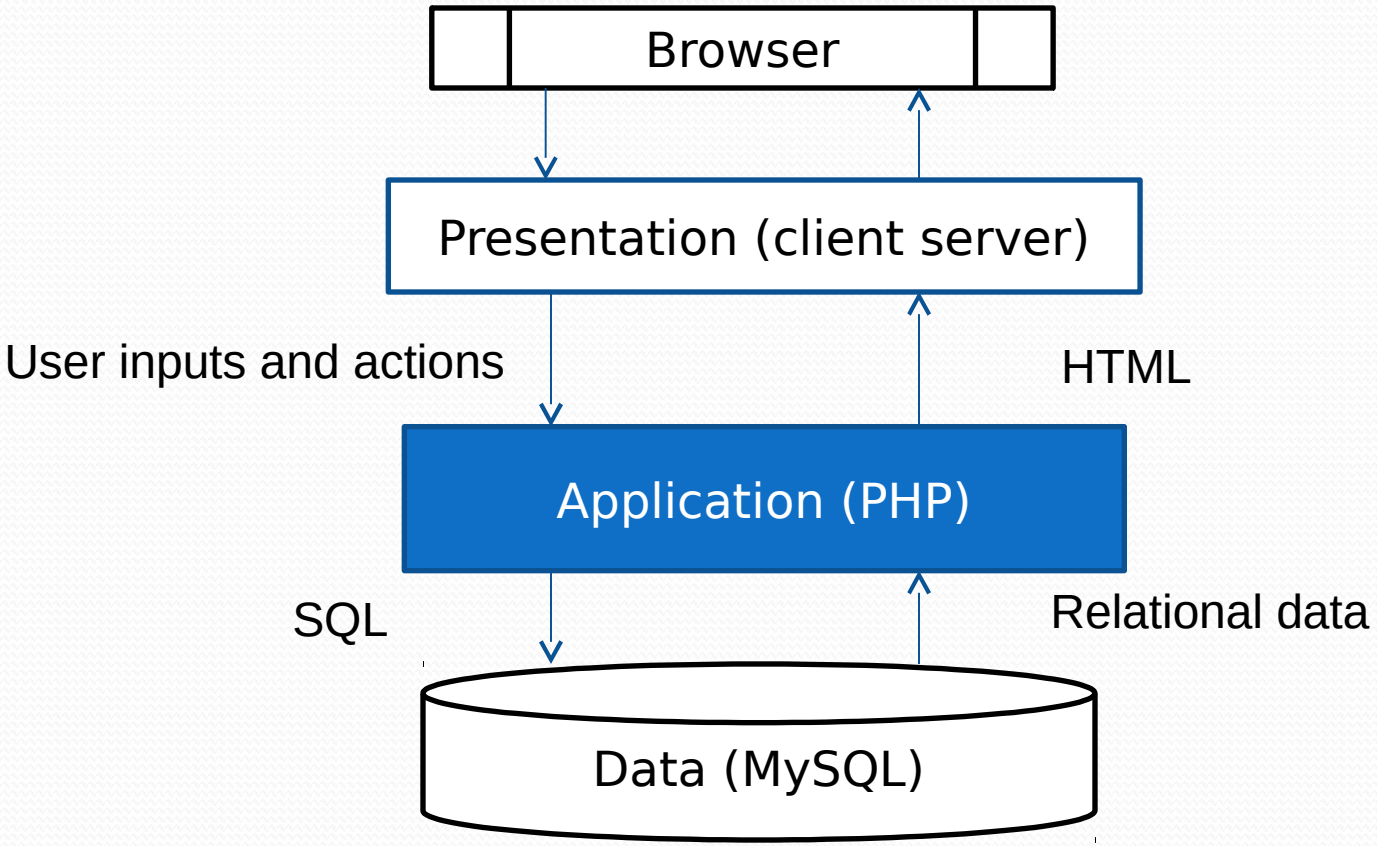
- Presentation tier
- Application tier
- Data tier

Web System Architecture

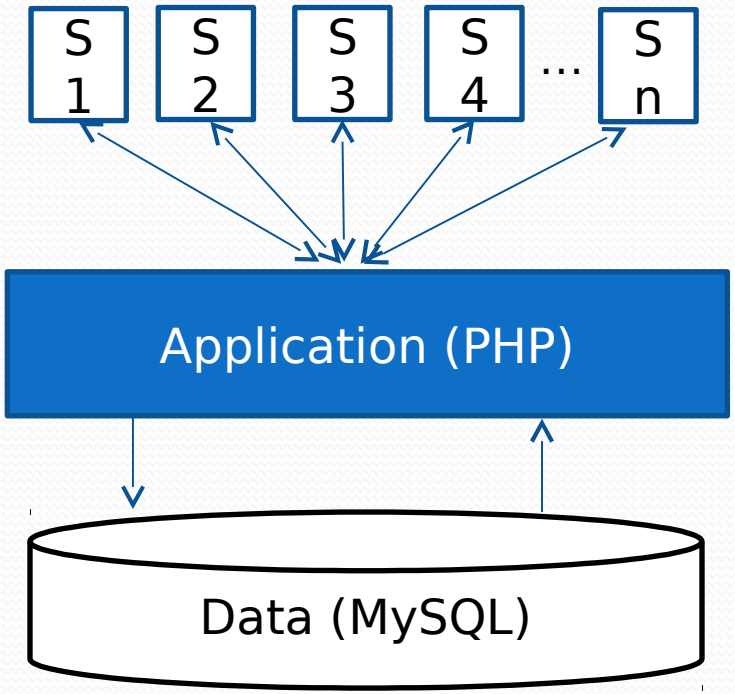
- Three-tier (or multi-tier) architecture

- Client server: presentation and user interaction
- Application server: application and business logic
- Database server: store the data and communicate with the application server

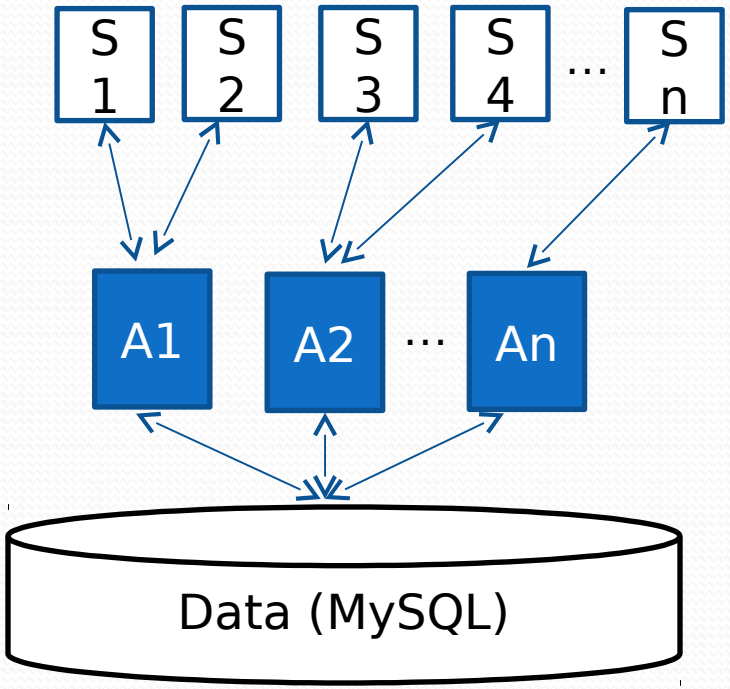
Three-Tier Architecture



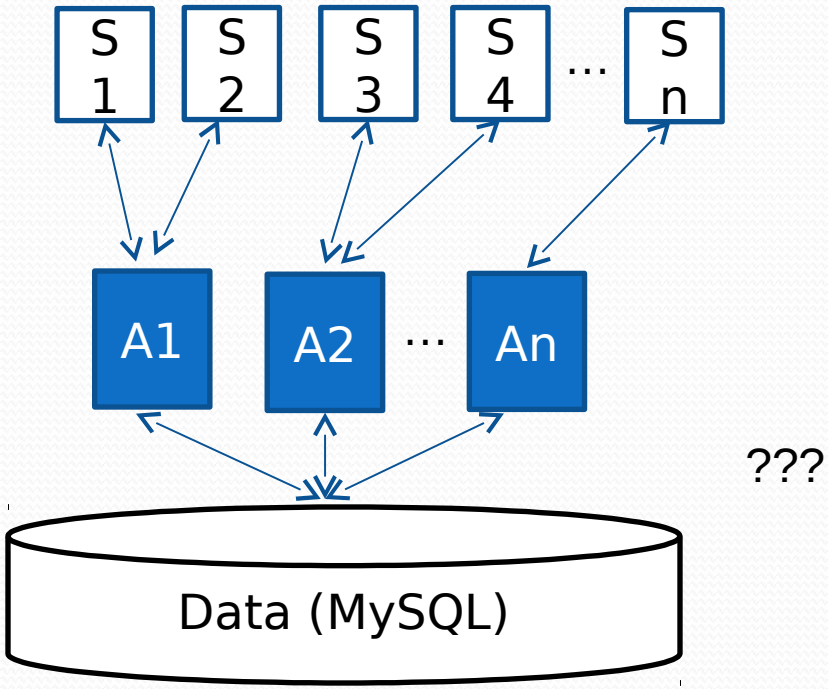
Scalability: Presentation Tier



Scalability: Application Tier



Scalability: Data Tier



???

Example: Texas Buck Registry

- Client server displays info to user and captures clicks and user input.
- App server implements actions (creating profile, posting deer, etc.) by issuing query updates to the database server.
- Database stores users, deer profiles, counties, etc.

Database Connectivity

- Embedded SQL
- Language-specific libraries
  - ADO.NET in C#
  - JDBC in Java
  - PHP
    - mysqli (was mysql til 4.1.3)
    - PDO (PHP Data Objects)

PHP: Hypertext Preprocessor (PHP)

- A scripting language for web development
  - Server-side language (executed by web server not client server)
  - Can retrieve data from a database
  - Can generate a web page dynamically when browser requests it (HTML = static)

PHP formatting

- Format:

```
<? php
... code ...
?>
```

- Example:

```
<? php
echo ‘Hello World!’;
?>
```

PHP Variables

- Variable names must begin with $.
- Variables do not need to be declared or typed.
- When a variable is assigned a value from a certain class, all class methods become available.

String Values

- Double quotes and single quotes have different interpretations in PHP.
- Single quotes mean literal string with no processing.
- Double quotes mean replace variable names inside the string with their values.

Examples

```
$race = 'Frank Lloyd Wright

$raceSentence = 'I ran the $
// value is 'I ran the $race.'

$raceSentence = “I ran the $
/* value is
'I ran the Frank Lloyd Wright 5K. ' */
```

PHP Arrays

- Numeric and Associative.

- Numeric array example:

$popStars = array(“Kesha”, “Beyonce”, “Katy”, “Taylor”)

// indexed 0,1,2 so:

// $popStars[0] is “Kesha”, $popStars[1] is “Beyonce”, etc.

PHP Arrays

- Assigning new values to numeric arrays:

$popStars[3] = “Rihanna”;

- Assigning new values to the end of arrays:

$popStars[] = “Halsey”;

Associative Arrays

- Elements of an associative array $array are pairs x => y, where x is a key string and y is a value.
- If x => y is an element of $array, then $array[x] is y.

- Example:

$swedishPopStars[‘OG’] = “Robyn”;

$swedishPopStars[‘Hipster’] = “Lykke Li”;

$swedishPopStars[‘CoolGirl’] = “Tove Lo”;

$swedishPopStars[‘WTF’] = “Angie”;

Associative Arrays Example

```
$mysqlParams = array(
    ‘host’ => ‘mpcs53001.cs.uchicago.edu’,
    ‘user’ => ‘zfreeman’,
    ‘password’ => ‘topSecretAmazingPassword’,
    ‘database’ => ‘zfreemanDB’);

// $mysqlParams[ ‘user’ ] is ‘zfreeman’
// $ mysqlParams[ ‘database’ ] is ‘zfreemanDB’
```

Form Values

- $_GET
- $_POST
- $_REQUEST

- Setting variables with Form values:
  - $username = $_REQUEST[‘username’];

MySQL Connection in PHP

- MySQL Improved Extension – mysqli

$myConnection = mysqli_connect($host, $username, $password, [$database])

- Standard practice is to define the values of the params in anothe file (typically a config file).
  - Reference with require (or include)

Error Handling in PHP

- Error handling is standard. Format for previous connection:

or die(“Could not connect: ” . mysqli_connect_error());

- Can use loose error message language for debugging.
- Need clear error messages for users.

Selecting Database

- If initial connect did not include the database or switching to a different database.

```
mysqli_select_db($myConnection,$database)
    or die("Could not select database: " .
$database);
```

- If you omit $myConnection, the last open connection will be used.

Queries in PHP

```
$query = ‘SELECT raceName, runnerName
FROM Registrations’;
$result = mysqli_query($query)
    or die(“Query $query failed: ” .
mysqli_error());
```

- mysqli_query takes a string argument and returns a result or generates an error.

Cursors in PHP

mysqli_fetch_row

mysqli_fetch_assoc

mysqli_fetch_array

- When applied to the result of a query they return the next tuple as a numeric array, associative array, or both.
- Return false after the last tuple.

Cursor Example

```
While ($tuple = mysqli_fetch_array($result))
{
    echo $tuple[0];
    echo $tuple['runnerName'];
}
```

Results recalibration with php

- Modify Results, so every race result Rita Jeptoo is listed in gets recalibrated.

```
$DroppedRunner = trim($_REQUEST[‘runnerToDrop']);
$raceToRecalibrateQuery = “??”;
$result = mysqli_query($raceToRecalibrateQuery)
    or die(“Query $query failed: “ .mysqli_error());
while ($tuple = mysqli_fetch_array($result)) {
    ??
}
```

Results recalibration with php

- Modify Results, so every runner after a dropped runner is moved down a place in the results.

```
$DroppedRunner = trim($_REQUEST[‘runnerToDrop']);
$raceToRecalibrateQuery = “SELECT DISTINCT race FROM Results WHERE runnerName = $DroppedRunner”;
$result = mysqli_query($raceToRecalibrateQuery)
    or die(“Query $query failed: “ .mysqli_error());
while ($tuple = mysqli_fetch_array($result)) {
    mysqli_query(CALL RecalibrateAllResultsByRunner($tuple[0], $DroppedRunner))
```

Error Handling in PHP (MySQL)

- For SELECT statements that should return values, can check:

```
if (!$result) {
    die(...error handling...)
}
```

  - Not useful for CREATE/INSERT/UPDATE/DELETE/DROP

Sessions with PHP

- Basic session management: let php handle most of the details.
- Start a session with: session_start()
  - Resume session if already started
- Store session data in php defined variable (associative array) $_SESSION
- To end session, clear all variables:
  - Unset($_SESSION[‘<variable>’]);

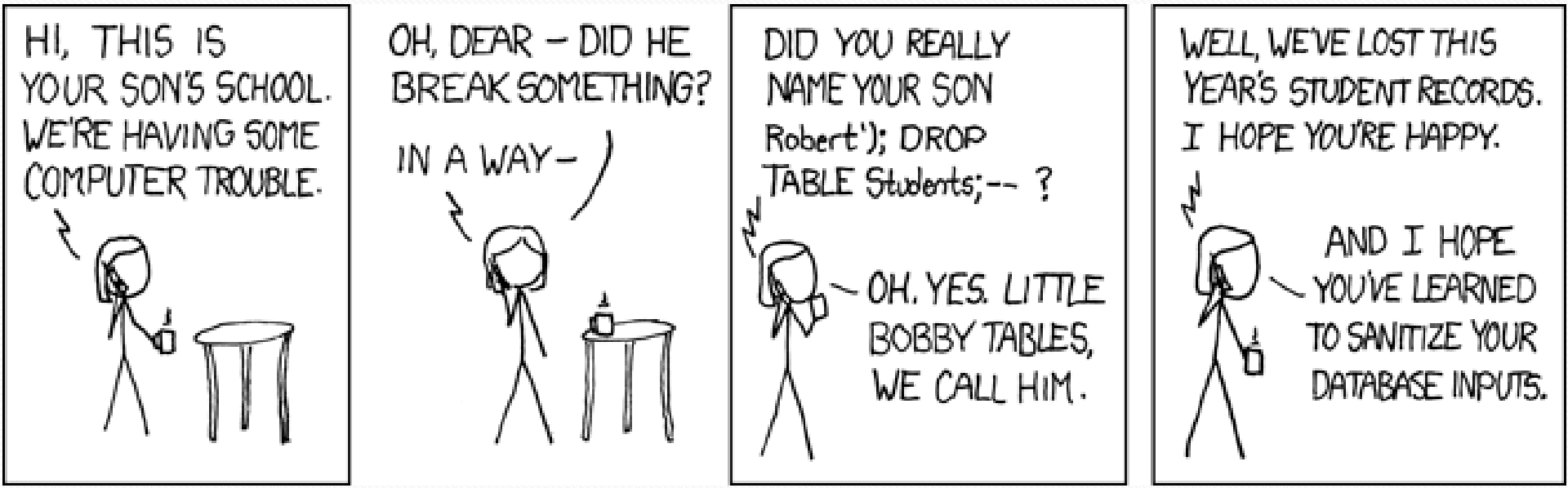
Session Example

```
<?php
session_start();
if (!isset($_SESSION[‘username’])) {
    print '<a href="/~login.html">Click to Login</a>';
}
else {
    …. other stuff …
}
?>
```

SQL Injection

- Hack input to change the query that gets executed.
- Usually includes ' followed by the hacker query followed by -- to treat anything after it as comments.
- Lesson: user input should not be trusted.

Little Bobby Tables



http://xkcd.com/327/

Bobby Tables Explanation

```
$newUserQ = 'INSERT INTO Students
    VALUES ('$newUserName');
```

- For Bobby, it would evaluate to:

```
INSERT INTO Students
    VALUES ('Robert'); DROP TABLE
Students ;--);
```

Second Example

$result=mysql_query("SELECT * FROM users WHERE username = $_REQUEST['username'] AND password = $_REQUEST['password']");

What if $_REQUEST['username'] = admin' OR 1=1 --

"SELECT * FROM users
WHERE username = 'admin' OR 1=1-- AND password = ''

What's the result of this query?


ZU 0666',0,0); DROP DATABASE TABLICE;--

Prepared Statements

- $sql = $myConnection->prepare("INSERT INTO Students (firstname, lastname, email) VALUES (:firstname, :lastname, :email)");
  $stmt->bindParam(':firstname', $firstname);
  $stmt->bindParam(':lastname', $lastname);
  $stmt->bindParam(':email', $email);
- $firstname = $_POST['firstname'];
  $lastname = $_POST['firstname'];
  $email = $_POST['firstname'];
  $stmt->execute();

Recap

- Web system architecture
- Database connectivity
- PHP
- SQL Injections

- Next class -THIS Saturday (11/18) and NEXT Tuesday (11/21):
  - Data warehousing (w/guest speaker Frank Greco from IBM)

Homework

- Assignment 8 (project and Gradiance lab)
  - Make sure your web account works!
  - Basic web programming: handle user inputs, generate SQL queries, run them using API (mysqli), collect results, display as HTML.
  - Assistance with PHP and web programming will be given during normal office hours this week.
  - Due AFTER Thansgiving

Questions?