CENG420

Server side
Lecture 3

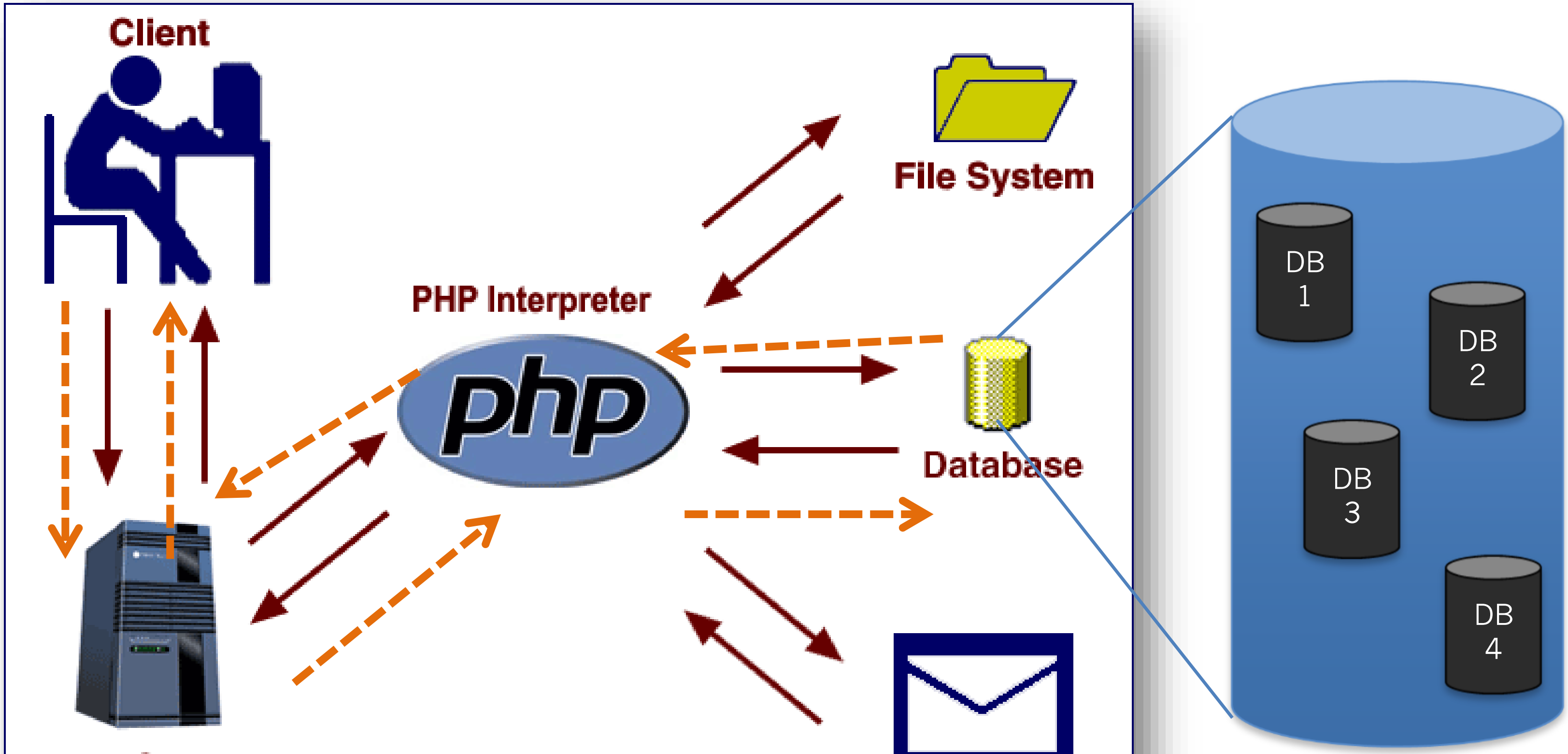
PHP
MYSQL integration

Fall 2019 LIU/BIU Dr. Ghantous

Why?

- Most websites use a database in the backend
- Databases usually hold data about:
  - Usernames and their login info
  - Website contents that might change by just changing the database contents
  - Any additional info the website might need

How they are really interconnected?



MYSQL (could be localhost or remote)

The process

- 1. Connect to MySQL.
- 2. Select the database to use.
- 3. Build a query string.
- 4. Perform the query.
- 5. Retrieve the results and output them to a web page.
- 6. Repeat Steps 3 through 5 until all desired data has been retrieved.
- 7. Disconnect from MySQL.

Connecting to Mysql

- All we need is:
- The **hostname** (where the mysql DB reside)
- The **username** and the **password** set by the administrator
- The database name

Returns an object representing the connection to the MySQL server

```
<?php
$con =
mysqli_connect("localhost","my_user","my_password","my_db");

// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
?>
```

Example

```
<?php

$db_hostname = 'localhost';
$db_database = ‘school';
$db_username = ‘root';
$db_password = ‘root’; // your username and passwords might be different on your xampp (usually username is “root”, password is “”)
$con = mysqli_connect($db_hostname,$db_username,$db_password,$db_database);

// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }

?>
```

What if your website contains several pages, each is trying to access the DB?

- Do we have to repeat this code over and over?
- We don’t have to if we write it ONCE, save it in a file, let’s say, called connection.php,
- And then just include it in any page that needs it
- We will use the keyword **`require_once`**
- Or you can use **`include`**

(code written before saved into connection.php)

```
<?php

$db_hostname = 'localhost';
$db_database = ‘school';
$db_username = ‘root';
$db_password = ‘root’;
$con = mysqli_connect($db_hostname,$db_username,$db_password,$db_database);
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
?>
```

Then from now on, in ***each*** page:

```
<?php
require_once ‘connection.php’ // or include
‘connection.php’
// all other php code
?>
```

Building and executing a query

- To execute queries, php uses **`mysqli_query(connection, StringQuery)`**
- **Example**:

```
<?php
require_once ‘connection.php’

$query = "SELECT * FROM classics";
$result = mysqli_query($con,$query);

if (!$result) die ("Database access failed: " .
mysqli_error());
?>
```

The result of the query is returned into the resource $result or FALSE if failure

$con is from the connection.php

Fetching Results

- From now on, you can use the resource returned by mysqli_query() to do many things.

- One of the things you can do is:

```
    $r = mysqli_num_rows($result)
```

- This will return the number of rows returned by the query

Fetching Results

- Another thing we can do is to fetch results:
- Using: $row = **mysqli_fetch_row($result)**
- This will return an array containing the current row, and then places the pointer on the next row.
- We can now access any attribute by: $row[0], $row[1],… and so on depending on the number of attributes

Example

```
<?php
require_once ‘connection.php’
$query = "SELECT * FROM classics";
$result = mysqli_query($con, $query);
$r = mysqli_num_rows($result)


for ($j = 0 ; $j < $r ; ++$j)
{
$row = mysqli_fetch_row($result);

echo 'Author: ' . $row[0] . '<br />';
echo 'Title: ' . $row[1] . '<br />';
echo 'Category: ' . $row[2] . '<br />';
echo 'Year: ' . $row[3] . '<br />';
echo 'ISBN: ' . $row[4] . '<br /><br />';
}
?>
```

Do we need to memorize the index of each attribute?

- No!
- We can use

  ```
  $row= mysqli_fetch_assoc($result)
  ```

- This will return an associative array so we can refer to attributes by their names instead of numbers. $row['title'] for example

Closing a connection

- **mysqli_close($con)**
- Note we use the $con to close the connection. → this was the handle returned by mysqli_connect

Additional features

- You can simply create, add, delete, update tables in your database from PHP

- Use same syntaxes as before, just change the query!!

The AUTO_INCREMENT field

- This is used by **mysql** when creating a table, usually with the ID attribute so the user doesn't need to worry about assigning IDs manually

- With the insertion of a new row, this field is incremented automatically

- PHP can access and read this value using: **mysqli_insert_id()**

Auto_increment (cont.)

- When inserting a new row, leave the auto_increment field NULL

```
$query = "INSERT INTO cats VALUES(NULL, 'Lynx', 'Stumpy', 5)";

$result = mysqli_query($con, $query);

echo "The Insert ID was: " . mysqli_insert_id();
```

Mysql_real_escape_string

- Use this when reading values inputted from users, before saving into database or using them into queries to avoid sql injection.

```
<?php
$user = mysql_fix_string($_POST['user']);
$pass = mysql_fix_string($_POST['pass']);
$query = "SELECT * FROM users WHERE user='$user' AND
pass='$pass'";

function mysql_fix_string($string)
{
if (get_magic_quotes_gpc()) $string = stripslashes($string);
return mysql_real_escape_string($string);
}
?>
```

Example

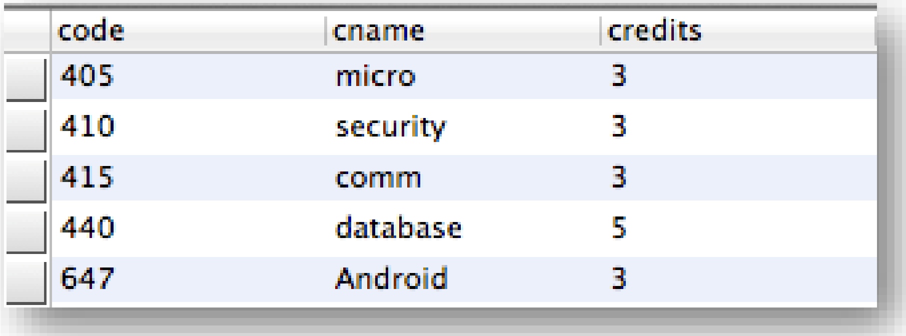
Practice example

- Create a single page that does several things with the database “SCHOOL”

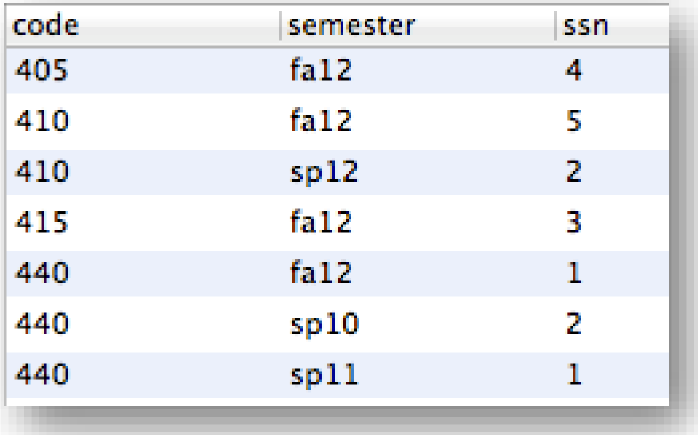
- School has 3 tables:

- PROFESSOR(ssn, pname, status, salary)
- COURSE(code, cname, credits)
- TAUGHT(code, semester, ssn)

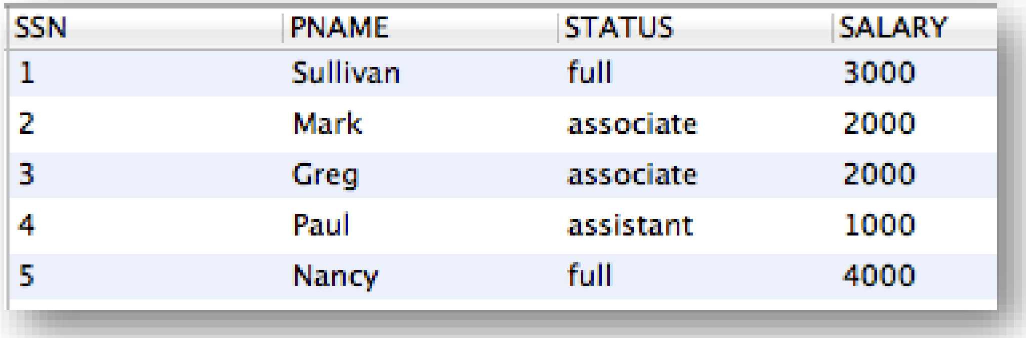
The table contents



code	cname	credits
405	micro	3
410	security	3
415	comm	3
440	database	5
647	Android	3



code	semester	ssn
405	fa12	4
410	fa12	5
410	sp12	2
415	fa12	3
440	fa12	1
440	sp10	2
440	sp11	1



SSN	PNAME	STATUS	SALARY
1	Sullivan	full	3000
2	Mark	associate	2000
3	Greg	associate	2000
4	Paul	assistant	1000
5	Nancy	full	4000

Create 2 pages till now

- Connection.php that establishes the connection to the db and can be included at the top of any other page:

connection.php

```
<?php
$db_hostname = 'localhost';
$db_database = ‘school';
$db_username = ‘root';
$db_password = ‘root’;
$con =
mysqli_connect($db_hostname,$db_username,$db_password,$db_database);
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
?>
```

The second page is used to display the form required to add the professor: db_example.php

- We have 2 options:
- Option 1: set the action of the form to a new page, let’s say add.php
- Option 2: set the action to the same page which is db_example.php

- Both will have similar programming!!

For now, we choose option 2. look next.

```
<?php

require_once 'connection.php';
if ( isset($_POST['ssn']) &&   isset($_POST['name']) && isset($_POST['status'])  &&
isset($_POST['salary'])  )
{
    $ssn= $_POST['ssn'];
    $name= $_POST['name'];
    $status= $_POST['status'];
    $salary= $_POST['salary'];
    // add values to the db to the table professor
 $sql_add_query = "INSERT INTO PROFESSOR VALUES('$ssn','$name','$status','$salary')";
     if(mysqli_query($con, $sql_add_query) === FALSE) die("Could not add the new professor");}
?>

<html>
    <body>
        <form class="s" method="POST" action="DB_example.php">
            <p>Insert a new professor</p>
            <table>
                <tr>
                    <td>SSN</td><td><input type="text" name="ssn"/></td></tr> <tr>
                    <td>Name</td><td><input type="text" name="name"/></td></tr>
<tr><td>Status</td><td><input type="text" name="status"/></td> </tr>
                <tr><td>Salary</td><td><input type="text" name="salary"/></td>  </tr>
<tr><td></td><td><input type="submit" value="Add professor"/></td></tr>

</table></form></body></html>
```

- change the index.php to display all the professors in a table in html. **Process**:
  - run a query to get all professors (select *)
  - Read the query results and organize them into a <table>

solution (some details omitted)

```
 <?php require_once ‘connection.php’;
 $query = "SELECT * FROM professors";
 $result = mysqli_query($con, $query);
 $r=  mysqli_num_rows($result); ?>


<table border="1">
<tr><td>ssn</td><td>name</td>
<td>status</td><td>salary</td><td>delete</td>
</tr>
<?php
for($j=0; $j<$r; $j++){
$fetched_row = mysqli_fetch_assoc($result);
$ssn = $fetched_row['ssn'];
$name = $fetched_row['name'];
$status = $fetched_row['status'];
$salary = $fetched_row['salary'];
echo "<tr><td>$ssn</td><td>$name</td><td>$status</td><td>$salary</td>"
. "<td><a href='delete.php?proffssn=$ssn'>delete</a></td>"                  .
"</tr>";
 }?>
</table>
```

what do you think is this?

- Assignment: design the page delete.php
  It deletes the professor and then redirects the user back to the index page

To redirect a user, use:

```
header("location:index.php");
```