EC201 – WEB TECHNOLOGIES
PHP and MySQL Integration

Querying Web Databases

- Connect to the server
  - Hostname of the database server
  - User name/ID and a password
- Select the database to work on
- Run the query on the database.
  - Replies back with a result set resource
- Retrieve a row of results
- Process the attribute values
- Close the connection to database server

Querying Web Databases with PHP

- Address Book database from the previous lab
  - Contacts, Groups

```
$conn = mysql_connect("localhost", "root", "");

mysql_select_db("addressbook", $conn);

$results = mysql_query("SELECT * FROM Contacts", $conn);

while ($row = mysql_fetch_array($result)) {
   print "$row['firstName'] $row['email']";
}
```

Handling Errors


```
$conn = mysql_connect("localhost", "root", "");
      or die("Could not connect: " . mysql_error());

mysql_select_db("addressbook", $conn);
      or die("Could not connect: " . mysql_error());

$results = mysql_query("SELECT * FROM Contacts", $conn);
      or die("Invalid query: " . mysql_error());

while ($row = mysql_fetch_array($result)) {
   print "$row['firstName'] $row['email']";
}
```

MySQL Functions (1/2)

- **mysql_connect** — Open a connection to a MySQL Server
- **mysql_ping** — Ping a server connection or reconnect if there is no connection
- **mysql_close** — Close MySQL connection
- **mysql_error** — Returns the text of the error message from previous MySQL operation
- **mysql_list_dbs** — List databases available on a MySQL server
- **mysql_select_db** — Select a MySQL database
- **mysql_list_tables** — List tables in a MySQL database
- **mysql_list_fields** — List MySQL table fields

MySQL Functions (2/2)

- **mysql_query** — Send a MySQL query
- **mysql_affected_rows** — Get number of affected rows in previous MySQL operation
- **mysql_num_rows** — Get number of rows in result
- **mysql_fetch_array** — Fetch a result row as an associative array, a numeric array, or both
- **mysql_fetch_assoc** — Fetch a result row as an associative array
- **mysql_fetch_row** — Get a result row as an enumerated array
- **mysql_fetch_object** — Fetch a result row as an object
- **mysql_fetch_field** — Get column information from a result

Adding Actionable Hyperlinks (1/2)


```
<table>
 <tr>
   <th>First Name</th> <th>Last Name</th> <th>Email</th>
   <th>Actions</th>
 </tr>
<?php
$conn = mysql_connect("localhost", "root", "");
mysql_select_db("AddressBook", $conn);
$result = mysql_query("select * from contacts");
while ($row = mysql_fetch_array($result)) {
  print "<td>" . $row["LastName"] . "</td>";
  print "<td>" . $row["email"] . "</td>";
  print "<td><a href='remove.php?id=".$row["ID"]."'>
            Remove </a></td>";
  print "</tr>";
}
mysql_close($conn);
?>
```

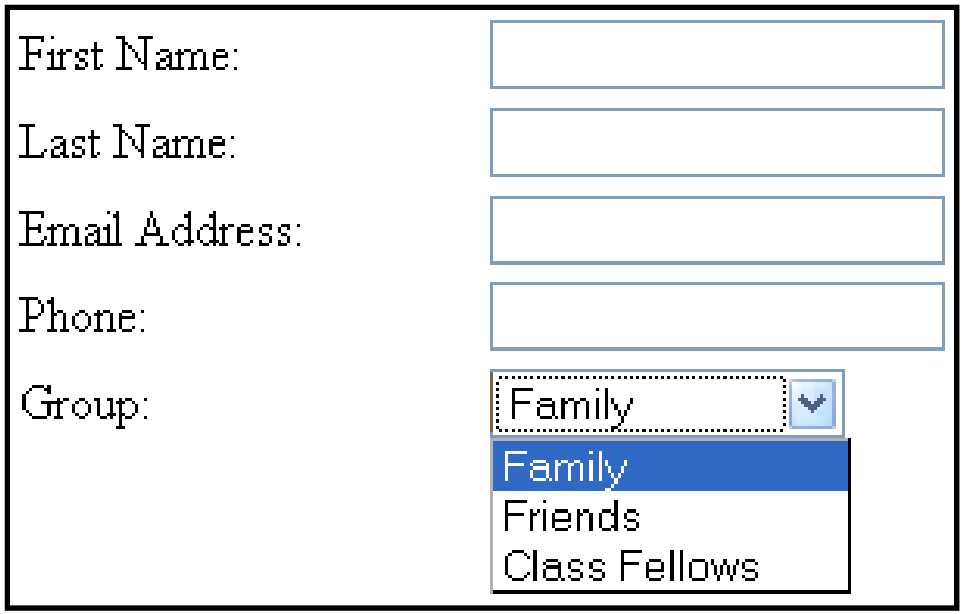
Adding Actionable Hyperlinks (2/2)


```
<?php
$cid = $_GET['id'];
$conn = mysql_connect("localhost", "root", "");
mysql_select_db("AddressBook", $conn);
$query = "DELETE FROM Contacts WHERE id={$cid}";
mysql_query($query);

If(mysql_affected_rows($query) ) {
  print mysql_affected_rows($query)." row(s) Deleted
       successfully";
}
?>
```

Interaction with the User

- Populate form fields with information from database
- Send the user input to PHP script for storing in the database



Populating Selection Box

- add-contact.php

```
<form action="do-add.php" method=post>
...
<?php
$results = mysql_query("SELECT * FROM Groups", $conn);

print '<select name="group">';

while ($group = mysql_fetch_array($results)) {
  print "<option value='" . $group["id"] . "'> " .
$group["name"]" . "</option>";
}

print '</select>';
?>
...
```

Storing User Input in Database

- do-add.php

```
...

$group_id = _POST['group'];

$query = "INSERT INTO Contacts VALUES … group=$group_id";

mysql_query($query, $conn);

if( mysql_affected_rows($conn) ) {
  print "New contact added successfully";
}

...
```

Secure Configuration

- Don’t run MySQL as **administrator/root**. Run it as a user created specifically for this purpose. Don’t use this account for anything else.
- Don’t access web database with **root** user. Create a separate admin account for each database for reading and writing from PHP script.
- Disallow access to port 3306 (or whatever port you have MySQL running on) except from trusted hosts.

Accounts and Privileges

- All MySQL accounts should have a password, especially **root**.
- Grant users the minimum level of privilege required to do their job.
  - Principle of Least Privilege
- Set permissions on the database directories so that only appropriate user can access them.
- Only the root user should have access to the **mysql** database, which contains privilege information.

Granting Privileges

- This is a command to create users and give them privileges. A simplified general syntax is
  - **GRANT *privileges* ON *item* TO *user_name***
  - **[IDENTIFIED BY '*password*'] [WITH GRANT OPTION]**
- If you use WITH GRANT OPTION, you allow the user to grant other users the privileges that you have given to him.
- REVOKE is opposite of GRANT.

Privileges

- **SELECT** allows users to select (read) records from tables.
- **INSERT** allows users to insert new rows into tables.
- **UPDATE** allows users to change values in existing table rows.
- **DELETE** allows users to delete table rows (records).
- **INDEX** allows user to index tables
- **ALTER** allows users to change the structure of the database.
  - Adding columns
  - Renaming columns or tables
  - Changing the data types of tables
- **DROP** allows users to delete databases or tables.
- **CREATE** allows users to create new databases or tables.
  - If a specific database or table is mentioned in the GRANT statement, users can only create that database or table.
- **USAGE** allows users nothing.
- **ALL** means just that.

Using Encryption

- Don’t store application passwords in plaintext in the database. (Use hashing mechanisms)
  - PHP has an in built **sha1()** function that calculates hashing scheme of a string.
  - The result of sha1() can not be translated back into the original string.
  - This makes it a good way to store password.
    - $safe_password=sha1($password);

Conclusion

- Learning Outcomes
  - Querying Web Databases in PHP
    - PHP MySQL Functions
  - User Interaction with Web Databases
  - Error Handling and Security Considerations
- Essential Reading
  - Web Database Application (2nd Ed.), Chapter 6
- What is Next?
    - Semester Project Proposal
  - Next Lab: Practicing PHP + MySQL