UNIT V: Using PHP with MySQL

5.1 Connecting to MySQL and Selecting the Database

The first step for Connecting to MySQL is to call mysqli_connect() function
The Syntax for this function is as follows:

$conn = mysqli_connect (hostname or servername, username, password, db_name);

The first three arguments sent to the function (hostname, username, and password) are based upon the users of MySQL database.

The hostname value will be localhost.
The fourth argument is the name of the database to use.
This is the equivalent of saying USE databasename within the mysql client.
If the connection was made, the $con variable, short for database connection, will become a reference point for all of your subsequent database interactions. Most of the PHP functions for working with MySQL will take this variable as its first argument.

Create a Connection to a MySQL Database

Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the mysqli_connect() function.

Servername – Specifies the server to connect to. Default value is "localhost"
Username - Specifies the username to log in with. Default value is the name of the user that owns the server process
Password - Specifies the password to log in with. Default is ""
DatabaseName- to provide database name created in PhpMyAdmin.

Example

```
<?php
$con = mysqli_connect("localhost","root","",”my_db”);
if (!$con)
{
Print 'Could not connect: ' . mysqli_error();
}
Else
{
Print 'Connect ‘;
}
// some code
?>
```

Closing a Connection

The connection will be closed automatically when the script ends. To close the connection before, use the mysqli_close() function:

Syntax : mysqli_close(connection variable);

Selecting Database:

PHP uses mysqli_select_db function to select the database on which queries are to be performed. This function takes two parameters and returns TRUE on success or FALSE on failure.

Syntax : mysqli_select_db (connection variable , string $dbname) : bool

```
Ex. $db = mysqli_select_db( $conn, 'stud' );
if(! $db) {
        die('Could not select database: ' . mysqli_error($conn));
}
```

5.2 Executing Simple Queries

Once you have successfully connected to and selected a database, you can start performing queries. These queries can be as basic as inserts, updates, and deletions or as involved as complex joins returning numerous rows.

In any case, the PHP function for executing a query is mysqli_query():

Syntax : mysqli_query(connection variable , string $query);

Ex. $result=mysqli_query($conn,$sql);

The mysqli_query() function takes the database connection as its argument and the query itself.

For simple queries like INSERT, UPDATE, DELETE, etc. (which do not return records), the
$r variable short for result will be either TRUE or FALSE, depending upon whether the query executed successfully.

Example

Inserting Record into Database

After a database and a table have been created, we can start adding data in them.

The SQL query must be quoted in PHP
String values inside the SQL query must be quoted
Numeric values must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table:

Syntax:
```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Example

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyMylist (firstname, lastname, email)
VALUES ('Shinde', 'Gopal', 'gopal@gmail.com')";
$sql = "DELETE FROM MyGuests WHERE id=3";
if (mysqli_query($conn, $sql)) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```
Multiple SQL statements must be executed with the **mysqli_multi_query()** function.

5.3 Retrieving Query Results

First, we set up an SQL query that selects the column name from table.
The SELECT statement is used to select data from one or more tables:

Syntax

SELECT column_name(s) FROM table_name

or we can use the * character to select ALL columns from a table

SELECT * FROM table_name
The next line of code runs the query and puts the resulting data into a variable called $result.

Then, the function num_rows() checks if there are more than zero rows returned.

If there are more than zero rows returned, the function fetch_assoc() puts all the results into an associative array that we can loop through.

The while() loop loops through the result set and outputs the data from the no of column name.

Example

```
<?php
// Create connection
$conn = mysqli_connect(“localhost”,”root”,”” ,”myDB”);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyList";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"].
"<br>";
  }
} else { echo "0 results";}
mysqli_close($conn);?>
```

5.4 Ensuring Secure SQL

Database security with respect to PHP comes down to three broad issues:
1. Protecting the MySQL access information
2. Not display too much about the database.
3. Being careful when running queries, particularly those involving user submitted data.

You can accomplish the first objective by securing the MySQL connection script outside of the Web directory so that it is never viewable through a Web browser.

The second objective is achieved by not letting the user see PHP's error messages or your queries (in these scripts, that information is printed out for your debugging purposes; you'd never want to do that on a live site).

For the third objective, there are numerous steps you can and should take, all based upon the premise of never trusting user supplied data.

First, validate that some value has been submitted, or that it is of the proper type (number, string, etc.).

Second, use regular expressions to make sure that submitted data matches what you would expect.

Third, you can typecast some values to guarantee that they're numbers.

Fourth recommendation is to run user submitted data through the mysql_real_ escape_string() function.

This function cleans data by escaping what could be problematic characters.

```
$clean = mysql_real_escape_string($dbc, data);
```

For security purposes, mysql_real_escape_string() should be used on every text input in a form.

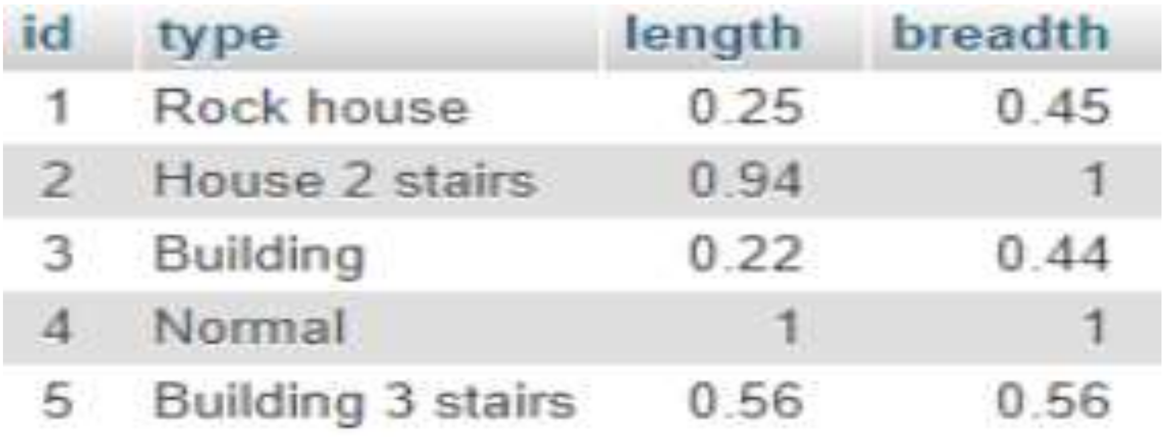
5.5 Counting Returned Records

We can get the total number of rows in a table by using the MySQL mysqli_num_rows() function.

Syntax:

```
mysqli_num_rows( result );
```

The result is to specify the result set identifier returned by mysqli_query() function.



id	type	length	breadth
1	Rock house	0.25	0.45
2	House 2 stairs	0.94	1
3	Building	0.22	0.44
4	Normal	1	1
5	Building 3 stairs	0.56	0.56

Example:

```
$sql = "SELECT * from building";

if ($result = mysqli_query($con, $sql)) {

   // Return the number of rows in result set
   $rowcount = mysqli_num_rows( $result );

   // Display result
   printf("Total rows in this table :  %d\n", $rowcount);
 }
```

Output:

```
Total rows in this table : 5
```

We count the table rows using MySQL count () function. It’s an aggregate function used to count rows.

Syntax: select count (*) from table;

5.6 Updating Records with PHP

The UPDATE statement is used to update existing records in a table

Syntax:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

Example:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE Mylist SET lastname='Ghodke' WHERE id=1";

if (mysqli_query($conn, $sql)) {
  echo "Record updated successfully";
}
else {
  echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

The End