

PHP – MySQL

DATABASE CONNECTIVITY

What is MySQL?

- A database system used on the web
- A database system that runs on a server
- Ideal for both small and large applications
- Very fast, reliable, and easy to use
- Uses standard SQL
- MySQL compiles on a number of platforms
- Free to download and use
- Developed, distributed, and supported by Oracle Corporation

PHP – MySQL system

- PHP combined with MySQL are cross-platform
 - (you can develop in Windows and serve on a Unix platform)
- Database Queries
 - A query is a question or a request.
 - We can query a database for specific information and have a recordset returned.
 - Look at the following query (using standard SQL):

```
SELECT LastName FROM Employees
```

The query above selects all the data in the "LastName" column from the "Employees" table.

- Install XAMPP or MAMP to use PHP – MySQL combination

PHP connection to MySQL

PHP 5 and later can work with a MySQL database using:

1. MySQLi extension (the "i" stands for improved)
 - i. Object – oriented
 - ii. Procedural
2. PDO (PHP Data Objects)

Earlier versions of PHP used the MySQL extension

PHP connection to MySQL DB Connect

- Connection to Server before
connecting to MySQL Database
- MySQLi Object Oriented Method

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

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";

// Close connection
$conn->close();
?>
```

Create Database DB Create

- A database consists of one or more tables.
- You will need special CREATE privileges to create or to delete a MySQL database.
- The CREATE DATABASE statement is used to create a database in MySQL.

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

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

Create Table

- A database table has its own unique name and consists of columns and rows
- The CREATE TABLE statement is used to create a table in MySQL.
- We will create a table named "MyGuests", with five columns:
 - id
 - firstname
 - lastname
 - email
 - reg_date

```
CREATE TABLE MyGuests (  
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  firstname VARCHAR(30) NOT NULL,  
  lastname VARCHAR(30) NOT NULL,  
  email VARCHAR(50),  
  reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
  CURRENT_TIMESTAMP  
)
```

Create Table

- The data type specifies what type of data the column can hold.
- After the data type, we can specify other optional attributes for each column:
 - NOT NULL - Each row must contain a value for that column, null values are not allowed
 - DEFAULT value - Set a default value that is added when no other value is passed
 - UNSIGNED - Used for number types, limits the stored data to positive numbers and zero
 - AUTO INCREMENT - MySQL automatically increases the value of the field by 1 each time a new record is added
 - PRIMARY KEY - Used to uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number, and is often used with AUTO_INCREMENT
- Each table should have a primary key column (in this case: the "id" column).
- Its value must be unique for each record in the table

Create Table

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

Insert data in Table

- After a database and a table have been created, we can start adding data in them.
- Here are some syntax rules to follow:
 1. The SQL query must be quoted in PHP
 2. String values inside the SQL query must be quoted
 3. Numeric values must not be quoted
 4. The word NULL must not be quoted

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

```
INSERT INTO table_name (column1, column2, column3,...)
```

```
VALUES (value1, value2, value3,...)
```

- If a column is AUTO_INCREMENT (like the "id" column) or TIMESTAMP with default update of current_timestamp (like the "reg_date" column), it is no need to be specified in the SQL query; MySQL will automatically add the value.

Insert data in Table

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Update data in Table

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

```
UPDATE table_name
```

```
SET column1=value, column2=value2,...
```

```
WHERE some_column=some_value
```

- The WHERE clause in the UPDATE syntax: The WHERE clause specifies which record or records that should be updated.
- If you omit the WHERE clause, all records will be updated!

Update data in Table

- The following examples update the record with id=2 in the "MyGuests" table

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

Retrieve data from Table

- The SELECT statement is used to select data from one or more tables:

```
SELECT column_name(s) FROM table_name
```

- We can use the * character to select ALL columns from a table:

```
SELECT * FROM table_name
```

Retrieve data from Table

- First, we set up an SQL query that selects the id, firstname and lastname columns from the MyGuests table.
- The next line of code runs the query and puts the resulting data into a variable called \$result.
- **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 id, firstname and lastname columns

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

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

Delete data from Table

- The DELETE statement is used to delete records from a table:

```
DELETE FROM table_name
```

```
WHERE some_column = some_value
```

- Notice the WHERE clause in the DELETE syntax: The WHERE clause specifies which record or records that should be deleted.
- If you omit the WHERE clause, all records will be deleted!

Delete data from Table

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

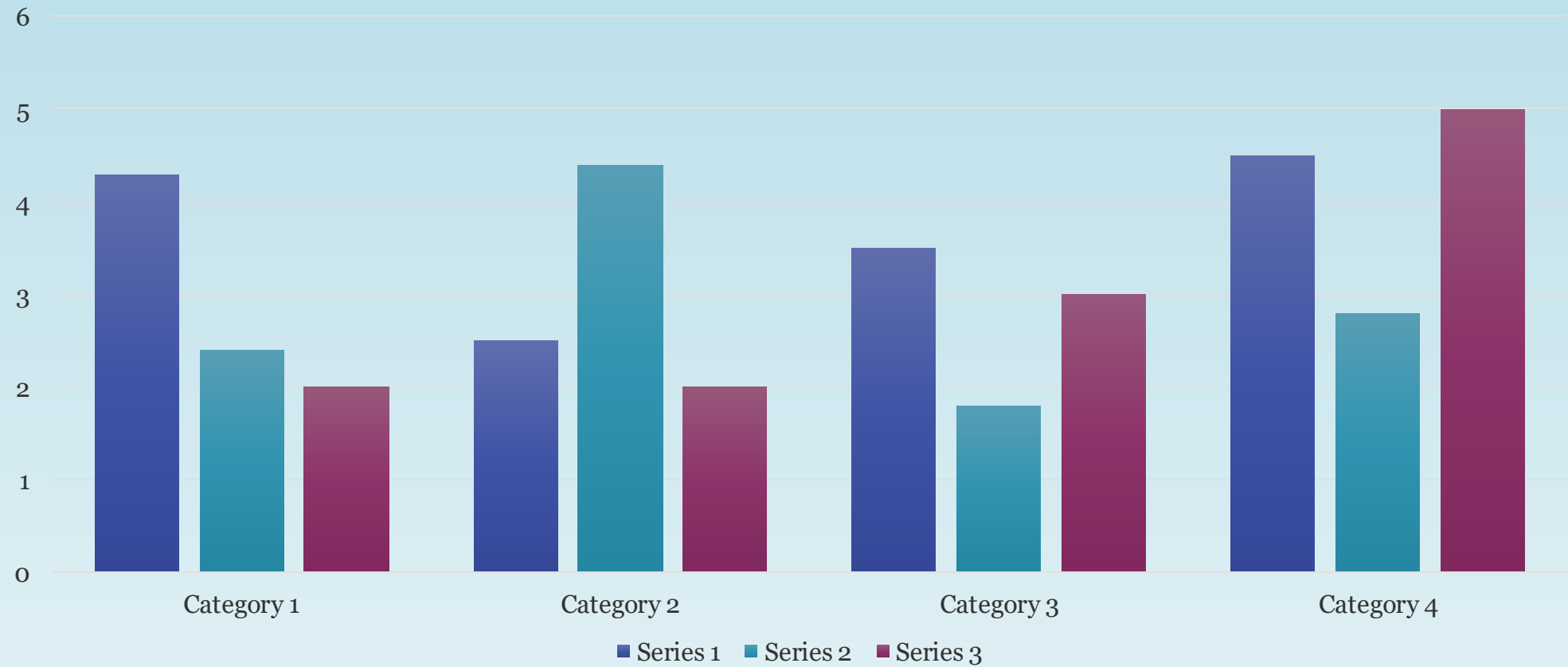
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}

$conn->close();
?>
```

Title and Content Layout with Chart





Picture with Caption Layout

Caption

Two Content Layout with Table

- First bullet point here
- Second bullet point here
- Third bullet point here

Class	Group A	Group B
Class 1	82	95
Class 2	76	88
Class 3	84	90

Two Content Layout with SmartArt

- First bullet point here
- Second bullet point here
- Third bullet point here

