PHP Connect to MySQL

PHP 5 onwards MySQL database using:

- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**

Should you use MySQLi or PDO?

A short answer, it would be **"Whatever you like".** Both MySQLi and PDO have their advantages:

PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases. So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries.

MySQLi, you will need to rewrite the entire code - queries included. Both are object-oriented, but **MySQLi** also **offers** a **procedural API**.

Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

MySQL Examples in Both MySQLi and PDO Syntax

- MySQLi (Object-Oriented)
- MySQLi (Procedural)
- PDO(PHP Data Objects)

MySQLi Installation: (Linux and Windows) http://php.net/manual/en/mysqli.installation.php

PDO Installation: http://php.net/manual/en/pdo.installation.php

To connect MySQL database, we need to be able to **connect to the server: Common for all types of connection**

```
<?php
$servername = "localhost";
$username = "root";
$password = "";  //For XAMPP no password

?>
```

PHP Connect to MySQL

MySQLi Object-Oriented	MySQLi Procedural	PDO
1.Create & Check connection		

<?php
$conn = new mysqli($servername, $username, $password);
 	
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
	<?php
	$conn = mysqli_connect($servername, $username, $password);
	
	if (!$conn) {
	die("Connection failed: " . mysqli_connect_error());
	}
	echo "Connected successfully";
	?>
		<?php
		try {
		$conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
		// set the PDO error mode to exception
		$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
		echo "Connected successfully";
		} catch(PDOException $e) {
		echo "Connection failed: ". $e->getMessage();
		} ?>

2. Close the Connection		
$conn->close();	mysqli_close($conn);	$conn = null;
3. Create a MySQL Database		

<?php
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
echo "Database created successfully";
} else {
echo "Error creating database: " . $conn->error;
}
?>
	<?php
	$sql = "CREATE DATABASE myDB";
	if (mysqli_query($conn, $sql)) {
	echo "Database created successfully";
	} else {
	echo "Error creating database: " . mysqli_error($conn);
	}
	?>
		<?php
		try {
		$conn = new PDO("mysql:host=$servername", $username, $password);
		// set the PDO error mode to exception
		$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
		$sql = "CREATE DATABASE myDB";
		// use exec() because no results are returned
		$conn->exec($sql);
		echo "Database created successfully
		";
		} catch(PDOException $e) {
		echo $sql . "
		" . $e->getMessage();
		}
		?>

PHP Connect to MySQL

4. Create a MySQL Table (Common Query [$sql] for all types)

```
<?php
$dbname = "myDB"; //Add to beginning before table creation
$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
)"; ?>
```

- 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

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

```
<?php
if (mysqli_query($conn, $sql)) {
  echo "Table MyGuests created
successfully";
} else {
  echo "Error creating table: " .
mysqli_error($conn);
}
?>
```

```
<?php
try {
  $conn
= new PDO("mysql:host=$servername;dbname
=$dbname", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
// use exec() because no results are
returned
  $conn->exec($sql);
  echo "Table MyGuests created
successfully";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
} ?>
```

PHP Connect to MySQL

5. Insert Data Into MySQL(Common Query [$sql] for all types)

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

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

```
<?php
if (mysqli_query($conn, $sql)) {
  echo "New record created
successfully";
} else {
  echo "Error: " . $sql . "<br>" .
mysqli_error($conn);
}
?>
```

```
<?php
try {
  $conn
= new PDO("mysql:host=$servername;dbname
=$dbname", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
   $conn->exec($sql);
  echo "New record created
successfully";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}?>
```

6. Get ID of the Last Inserted Record

//To perform an INSERT or UPDATE on a table with an AUTO_INCREMENT field, we can get the ID of the last inserted/updated record immediately. In the table "MyGuests", the "id" column is an AUTO_INCREMENT field:

```
<?php
if ($conn->query($sql) === TRUE) {
  $last_id = $conn->insert_id;
  echo "New record created
successfully. Last inserted ID is:
" . $last_id;
} else {
  echo "Error: " . $sql . "<br>" .
$conn->error;
}
?>
```

```
<?php
if (mysqli_query($conn, $sql)) {
  $last_id =
mysqli_insert_id($conn);
  echo "New record created
successfully. Last inserted ID is:
" . $last_id;
} else {
  echo "Error: " . $sql . "<br>" .
mysqli_error($conn);
}
?>
```

```
<?php
try {  $conn
= new PDO("mysql:host=$servername;dbname
=$dbname", $username, $password);
  $conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);  $conn-
>exec($sql);
  $last_id = $conn->lastInsertId();
  echo "New record created successfully.
Last inserted ID is: " . $last_id;
} catch(PDOException $e) {  echo $sql
. "<br>" . $e->getMessage();}
?>
```

PHP Connect to MySQL
