PHP and Database Programming

PHP is a popular server-side scripting language widely used for web development. One of its key strengths is its ability to interact with databases, enabling dynamic and data-driven websites.

# Connecting PHP to a Database

To connect PHP to a database, developers commonly use two methods:  
  
1. MySQLi (MySQL Improved): Provides an interface to interact with MySQL databases. It supports both procedural and object-oriented programming styles.  
Example:  
  
$servername = 'localhost';  
$username = 'username';  
$password = 'password';  
$dbname = 'myDB';  
  
$conn = new mysqli($servername, $username, $password, $dbname);  
if ($conn->connect\_error) {  
 die('Connection failed: ' . $conn->connect\_error);  
}  
  
2. PDO (PHP Data Objects): A database access layer providing a uniform method of access to multiple databases. It supports prepared statements and transactions.  
Example:  
  
$servername = 'localhost';  
$username = 'username';  
$password = 'password';  
$dbname = 'myDB';  
  
try {  
 $conn = new PDO('mysql:host=$servername;dbname=$dbname', $username, $password);  
 $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  
} catch (PDOException $e) {  
 echo 'Connection failed: ' . $e->getMessage();  
}

# Performing Database Operations

## Inserting Data

$sql = 'INSERT INTO users (username, email) VALUES ('JohnDoe', 'john@example.com')';  
if ($conn->query($sql) === TRUE) {  
 echo 'New record created successfully';  
} else {  
 echo 'Error: ' . $sql . '<br>' . $conn->error;  
}

## Retrieving Data

$sql = 'SELECT id, username, email FROM users';  
$result = $conn->query($sql);  
if ($result->num\_rows > 0) {  
 while($row = $result->fetch\_assoc()) {  
 echo 'id: ' . $row['id'] . ' - Name: ' . $row['username'] . ' - Email: ' . $row['email'] . '<br>';  
 }  
} else {  
 echo '0 results';  
}

## Updating Data

$sql = 'UPDATE users SET email='newemail@example.com' WHERE id=1';  
if ($conn->query($sql) === TRUE) {  
 echo 'Record updated successfully';  
} else {  
 echo 'Error: ' . $sql . '<br>' . $conn->error;  
}

## Deleting Data

$sql = 'DELETE FROM users WHERE id=1';  
if ($conn->query($sql) === TRUE) {  
 echo 'Record deleted successfully';  
} else {  
 echo 'Error: ' . $sql . '<br>' . $conn->error;  
}

# Security Considerations

When interacting with databases, it's crucial to prevent SQL injection attacks. Using prepared statements with bound parameters is a recommended practice:  
  
$stmt = $conn->prepare('INSERT INTO users (username, email) VALUES (?, ?)');  
$stmt->bind\_param('ss', $username, $email);  
$username = 'JaneDoe';  
$email = 'jane@example.com';  
$stmt->execute();

# Conclusion

PHP's integration with databases like MySQL allows developers to create dynamic, data-driven websites. By understanding how to connect to a database and perform basic operations securely, developers can build robust applications that interact with data efficiently.