

Dưới đây là một ví dụ về cách sử dụng **PDO** (PHP Data Objects) để kết nối với cơ sở dữ liệu trong PHP.

⇒ Ví dụ này sẽ trình bày cách kết nối đến một cơ sở dữ liệu MySQL và thực hiện một thao tác CRUD đơn giản (tạo mới dữ liệu).

## 1. Kết nối PDO với cơ sở dữ liệu MySQL

```
<?php
// Thông tin kết nối
$host = 'localhost'; // Địa chỉ máy chủ cơ sở dữ liệu
$dbname = 'test_db'; // Tên cơ sở dữ liệu
$username = 'root'; // Tên người dùng
$password = ''; // Mật khẩu
try {
    // Tạo kết nối PDO
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username,
$password);

    // Thiết lập chế độ lỗi
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo "Kết nối thành công!"; } catch (PDOException $e) {
    echo "Lỗi kết nối: " . $e->getMessage();
}
?>
```

## 2. Thực hiện thao tác Create (Thêm dữ liệu mới)

```
<?php
try {
    // Tạo câu lệnh SQL để thêm dữ liệu
    $sql = "INSERT INTO users (name, email) VALUES (:name, :email)";

    // Chuẩn bị câu lệnh
    $stmt = $pdo->prepare($sql);

    // Gán giá trị cho các tham số
    $name = 'John Doe';
    $email = 'johndoe@example.com';

    // Liên kết tham số với giá trị và thực thi câu lệnh
```

```

$stmt->bindParam(':name', $name);
$stmt->bindParam(':email', $email);           // Thực thi câu lệnh
$stmt->execute();

echo "Dữ liệu đã được thêm vào thành công!";
} catch (PDOException $e) {
    echo "Lỗi: " . $e->getMessage();
}

```

### 3. Thực hiện thao tác Read (Lấy dữ liệu)

```

<?php
try {
    // Câu lệnh SQL để lấy tất cả người dùng
    $sql = "SELECT * FROM users";

    // Chuẩn bị và thực thi câu lệnh
    $stmt = $pdo->query($sql);

    // Lấy tất cả kết quả dưới dạng mảng
    $users = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // Hiển thị dữ liệu
    foreach ($users as $user) {
        echo "ID: " . $user['id'] . " - Name: " . $user['name'] . " - Email: "
        . $user['email'] . "<br>";
    }
} catch (PDOException $e) {
    echo "Lỗi: " . $e->getMessage();
}

```

### 4. Thực hiện thao tác Update (Cập nhật dữ liệu)

```

<?php
try {
    // Câu lệnh SQL để cập nhật dữ liệu
    $sql = "UPDATE users SET name = :name, email = :email WHERE id = :id";

    // Chuẩn bị câu lệnh
    $stmt = $pdo->prepare($sql);

    // Gán giá trị cho các tham số
    $name = 'Jane Doe';

```

```

$email = 'janedoe@example.com';
$id = 1; // Giả sử ID là 1

// Liên kết tham số với giá trị và thực thi câu lệnh
$stmt->bindParam(':name', $name);
$stmt->bindParam(':email', $email);
$stmt->bindParam(':id', $id);

// Thực thi câu lệnh
$stmt->execute();

echo "Dữ liệu đã được cập nhật!";
} catch (PDOException $e) {
    echo "Lỗi: " . $e->getMessage();
}

```

## 5. Thực hiện thao tác Delete (Xóa dữ liệu)

```

<?php
try {
    // Câu lệnh SQL để xóa dữ liệu
    $sql = "DELETE FROM users WHERE id = :id";

    // Chuẩn bị câu lệnh
    $stmt = $pdo->prepare($sql);

    // Gán giá trị cho tham số
    $id = 1; // Giả sử ID là 1

    // Liên kết tham số với giá trị và thực thi câu lệnh
    $stmt->bindParam(':id', $id);

    // Thực thi câu lệnh
    $stmt->execute();

    echo "Dữ liệu đã được xóa!";
} catch (PDOException $e) {
    echo "Lỗi: " . $e->getMessage();
}

```

### Giải thích:

1. **Kết nối PDO:** Sử dụng đối tượng `PDO` để kết nối đến cơ sở dữ liệu với các thông tin như máy chủ, tên cơ sở dữ liệu, tên người dùng và mật khẩu.
2. **CRUD Operations:** Sử dụng các câu lệnh SQL tương ứng để thực hiện các thao tác `Create`, `Read`, `Update`, và `Delete` trên cơ sở dữ liệu.
3. **Xử lý lỗi:** Đặt chế độ báo lỗi cho PDO để dễ dàng xử lý và báo cáo lỗi khi kết nối hoặc thực thi các câu lệnh.

Lưu ý: Trước khi chạy các mã này, bạn cần có cơ sở dữ liệu MySQL và bảng `users` với các cột `id`, `name`, và `email` trong cơ sở dữ liệu của bạn.