

## Vì sao sử dụng PDO nên thực hiện 4 bước: prepare - bindParam - execute - fetch?

=> Đây là nền tảng quan trọng khi làm việc với **PDO (PHP Data Objects)** trong PHP để truy vấn cơ sở dữ liệu một cách an toàn và hiệu quả. 4 bước như sau 📌

---

### 🌸 1. `prepare()` — Chuẩn bị câu truy vấn

#### Mục đích:

- Tạo một **statement object** (câu truy vấn được biên dịch trước).
- Giúp PDO gửi **cấu trúc câu lệnh SQL** đến máy chủ CSDL **một lần**, không kèm dữ liệu.
- CSDL sẽ kiểm tra cú pháp, lập kế hoạch thực thi (execution plan) trước khi chạy.

#### Lợi ích:

- Giảm chi phí xử lý khi chạy nhiều lần cùng một câu truy vấn (chỉ khác dữ liệu).
- Tránh lỗi cú pháp SQL do dữ liệu đầu vào không an toàn.

#### Ví dụ:

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");
```

---

### 🔗 2. `bindParam()` hoặc `bindValue()` — Gắn giá trị vào tham số

#### Mục đích:

- Thay thế các **placeholder** (ví dụ `:email`) bằng **giá trị thực**.
- PDO sẽ tự động **escape** dữ liệu, giúp **chống SQL Injection**.

#### Khác biệt nhỏ:

- `bindParam()` → gắn **biến** (tham chiếu), giá trị được lấy **khi thực thi** (`execute()`).
- `bindValue()` → gắn **giá trị ngay lập tức**.

#### Ví dụ:

```
$email = "user@example.com";  
$stmt->bindParam(":email", $email, PDO::PARAM_STR);
```

---

### 3. `execute()` — Thực thi câu truy vấn

#### Mục đích:

- Gửi truy vấn (đã gắn tham số) lên CSDL để thực thi.
- Trả về `true` nếu thành công, `false` nếu lỗi.

#### Ví dụ:

```
$stmt->execute();
```

---

### 4. `fetch()` hoặc `fetchAll()` — Lấy kết quả

#### Mục đích:

- Lấy dữ liệu trả về từ truy vấn (SELECT).
- Có thể lấy từng dòng (`fetch()`) hoặc tất cả (`fetchAll()`).

#### Ví dụ:

```
$result = $stmt->fetch(PDO::FETCH_ASSOC);
```

---

### Tổng kết lý do nên dùng đủ 4 bước:

Bước	Mục đích chính	Lợi ích
<code>prepare()</code>	Chuẩn bị và biên dịch trước câu lệnh SQL	Tối ưu hiệu năng, chống lỗi cú pháp
<code>bindParam()</code>	Gắn giá trị an toàn vào truy vấn	Tránh SQL Injection
<code>execute()</code>	Thực thi câu truy vấn	Chạy truy vấn với dữ liệu đã gắn
<code>fetch()</code>	Lấy dữ liệu kết quả	Truy xuất dữ liệu dễ dàng, linh hoạt

---

### Ghi nhớ nhanh:

**PDO = Prepare → Bind → Execute → Fetch**

Giúp code PHP an toàn, gọn gàng, hiệu suất cao và dễ bảo trì.