**Các phương thức cơ bản làm việc với CSDL sử dụng lớp PDO**

Đầu tiên chúng ta phải biết cú pháp thực thi hàm của một đối tượng. Nó có dạng tổng quát như sau: $ten\_doi\_tuong->ten\_phuong\_thuc(danh\_sach\_doi\_so);

# Lệnh INSERT, UPDATE, DELETE.

Thêm mới (**insert**) và cập nhật (**update**) dữ liệu là những hoạt động cơ bản khi thao tác với database. Với PDO, mỗi hoạt động insert hay update được thực hiện qua 3 quá trình sử dụng cơ chế **Prepared Statement**

1. Prepare statement: Chuẩn bị một câu lệnh SQL làm khung/mẫu được gọi là **Prepared Statement** với các **Placeholder** (có thể hiểu placeholder đóng vai trò như tham số của các phương thức khi bạn khai báo hàm)
2. Bind params: Gắn giá trị thực vào các placeholder (tương tự như khi bạn truyền giá trị vào các tham số của phương thức)
3. Execute: Thực thi câu lệnh.

Có 2 loại Placeholder trong Prepared Statement là **Placeholder không định danh** (**Unnamed Placeholder**) và **Placeholder định danh** (**Named Placeholder**) như ví dụ sau:

$query = 'INSERT INTO users (name, email, age) values (?, ?, ?)';

$user = $conn->prepare($query);

//Hoặc

$query = 'INSERT INTO users (name, email, age) values (:name, :mail, :age)';

$user = $conn->prepare($query);

Dòng lệnh thứ nhất sử dụng Placeholder không định danh là các dấu hỏi - ?. Dòng lệnh thứ 2 sử dụng Placeholder định danh: :name, :mail, :age (lưu ý dấu hai chấm và placeholder không nhất thiết phải giống tên column). Sau đây là toàn bộ quá trình Insert và Update sử dụng 2 loại Placeholder nêu trên.

Ví dụ:

//Khởi tạo Prepared Statement từ biến $conn ở phần trước

$stmt = $conn->prepare('INSERT INTO users (name, email, age) values (?, ?, ?)');

//Gán các biến (lúc này chưa mang giá trị) vào các placeholder theo thứ tự tương ứng

$stmt->bindParam(1, $name);

$stmt->bindParam(2, $mail);

$stmt->bindParam(3, $age);

//Gán giá trị và thực thi

$name = "Vu Hoang Lam"<

$mail = "lamvh@live.com";

$age = 22;

$stmt->execute();

//Gán những giá trị khác và tiếp tục thực thi

$name = "Nguyen Van A";

$mail = "nva@live.com";

$age = 23;

$stmt->execute();

Đối với Named Placeholder, cách thực hiện cũng khá tương đồng với Unnamed Placeholder, chỉ khác là ta không dùng thứ tự placeholder để gán giá trị (bind) mà dùng chính tên của placeholder:

//Khởi tạo Prepared Statement từ biến $conn ở phần trước

$stmt = $conn->prepare('INSERT INTO users (name, email, age) values (:name, :mail, :age)');

//Gán các biến (lúc này chưa mang giá trị) vào các placeholder theo tên của chúng

$stmt->bindParam(':name', $name);

$stmt->bindParam(':mail', $mail);

$stmt->bindParam(':age', $age);

//Gán giá trị và thực thi

$name = "Vu Hoang Lam";

$mail = "lamvh@live.com";

$age = 22;

$stmt->execute();

# Đọc dữ liệu trả về (SELECT).

Khi *đọc* dữ liệu từ database, **PDO** sẽ trả về dữ liệu theo cấu trúc mảng (array) hoặc đối tượng (object) thông qua phương thức **fetch()**. Bạn nên thiết lập trước cấu trúc dữ liệu trước khi gọi phương thức này, **PDO** hỗ trợ các tuỳ chọn sau:

* **PDO::FETCH\_ASSOC**: Trả về dữ liệu dạng mảng với key là tên của column (column của các table trong database)
* **PDO::FETCH\_BOTH** (default): Trả về dữ liệu dạng mảng với key là tên của column và cả số thứ tự của column
* **PDO::FETCH\_BOUND**: Gán giá trị của từng column cho từng biến đã khởi tạo trước đó qua phương thức bindColumn()
* **PDO::FETCH\_CLASS**: Gán giá trị của từng column cho từng thuộc tính (property/attribute) của một lớp Class theo tên column và tên thuộc tính.
* **PDO::FETCH\_INTO**: Gán giá trị của từng column cho từng thuộc tính của một Class Instance (thể hiện của một lớp)
* **PDO::FETCH\_LAZY**: Gộp chung PDO::FETCH\_BOTH/PDO::FETCH\_OBJ
* **PDO::FETCH\_NU**M: Trả về dữ liệu dạng mảng với key là số thứ tự của column
* **PDO::FETCH\_OBJ**: Trả về một [Object của stdClass(link is external)](http://www.php.net/manual/en/language.types.object.php) với tên thuộc tính của Object là tên của column.

Trong thực tế, chúng ta chỉ thường dùng 3 kiểu fetch đó là: FETCH\_ASSOC, FETCH\_CLASS và FETCH\_OBJ. Để thiết lập cấu trúc dữ liệu (**Fetch Style** hay **Fetch Mode**) trước khi fetch ta dùng câu lệnh sau:

Chúng ta thiết lập kiểu fetch ngay khi gọi hàm fetch();

$stmt->fetch(PDO::FETCH\_ASSOC);