

# *Kết hợp PHP & MySQL*

---

# Nội dung

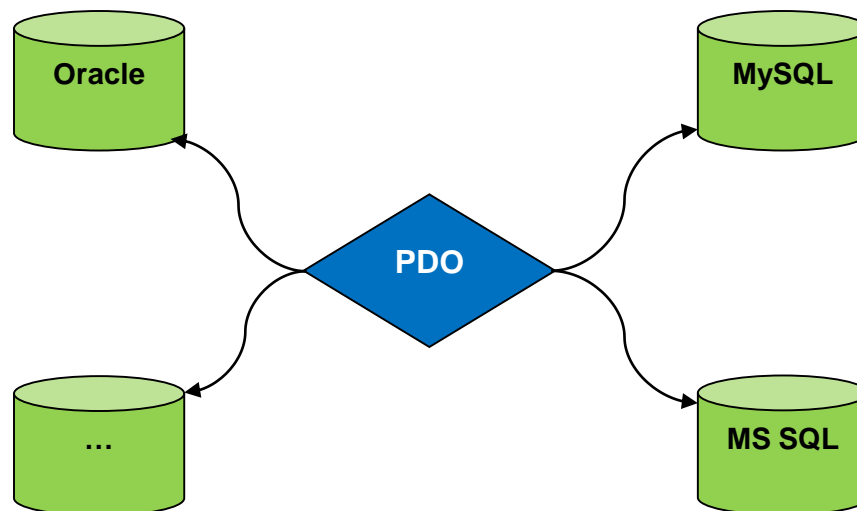
1. Tổng quan về PDO
2. Transaction
3. PDO Statement
4. Hiển thị dữ liệu
5. Thêm – xóa – cập nhật dữ liệu

# 1. Tổng quan về PDO

- ▶ PDO là gì?
- ▶ Vì sao chọn PDO?
- ▶ Tạo kết nối – đóng kết nối
- ▶ Thực thi lệnh SQL

# 1. Tổng quan về PDO > PDO là gì?

- ▶ PDO (PHP Data Object): là một lớp truy cập cơ sở dữ liệu cung cấp cho chúng ta các tiện ích để truy cập đến nhiều loại cơ sở dữ liệu



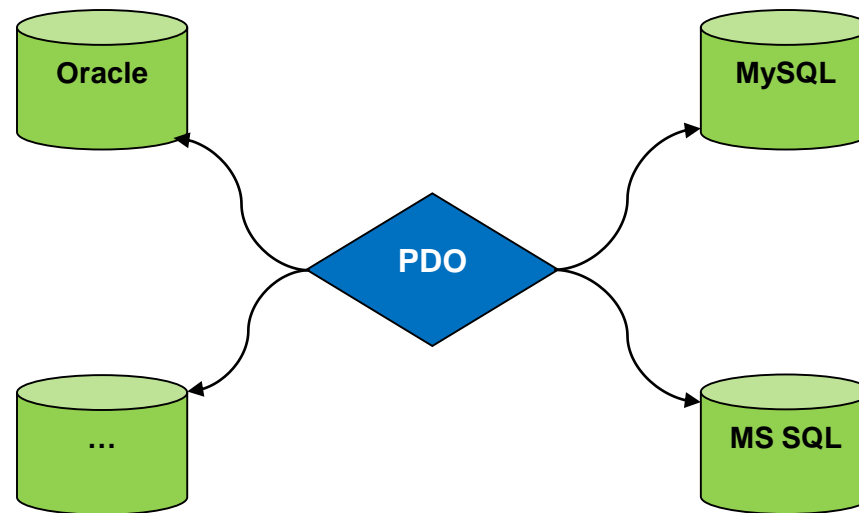
Mô hình làm việc giữa PDO và các loại CSDL

# 1. Tổng quan về PDO

- ▶ PDO là gì?
- ▶ **Vì sao chọn PDO?**
- ▶ Tạo kết nối – đóng kết nối
- ▶ Thực hiện lệnh SQL

# 1. Tổng quan về PDO > Vì sao chọn PDO?

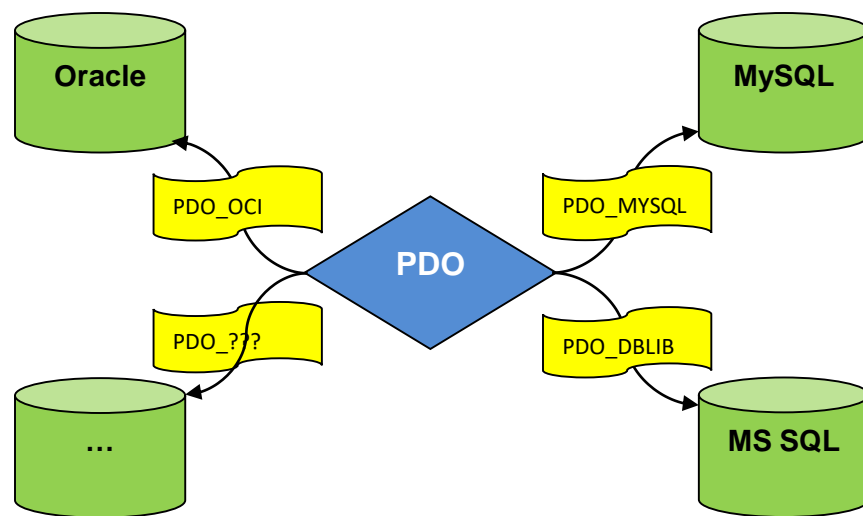
- Hướng đối tượng: rất phù hợp để chúng ta tích hợp vào mô hình MVC (Rất nhiều framework lớn hiện nay đều sử dụng PDO)



Mô hình làm việc giữa PDO và các loại CSDL

# 1. Tổng quan về PDO > Vì sao chọn PDO?

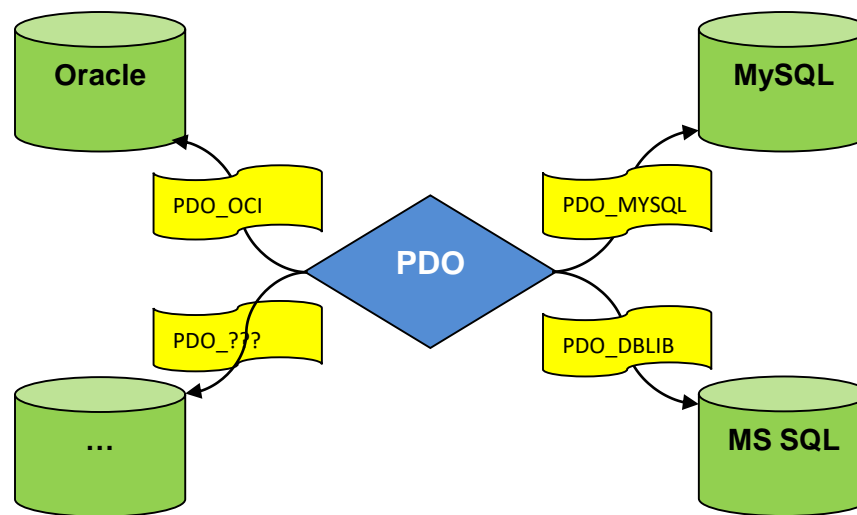
- ▶ Nhanh & đơn giản: việc thao tác với từng loại csdl sẽ do từng driver tương ứng đảm nhiệm



Mô hình làm việc giữa PDO và các loại CSDL

# 1. Tổng quan về PDO > Vì sao chọn PDO?

- Linh động: để chuyển đổi sang CSDL khác ta chỉ việc thay đổi tên driver tương ứng.

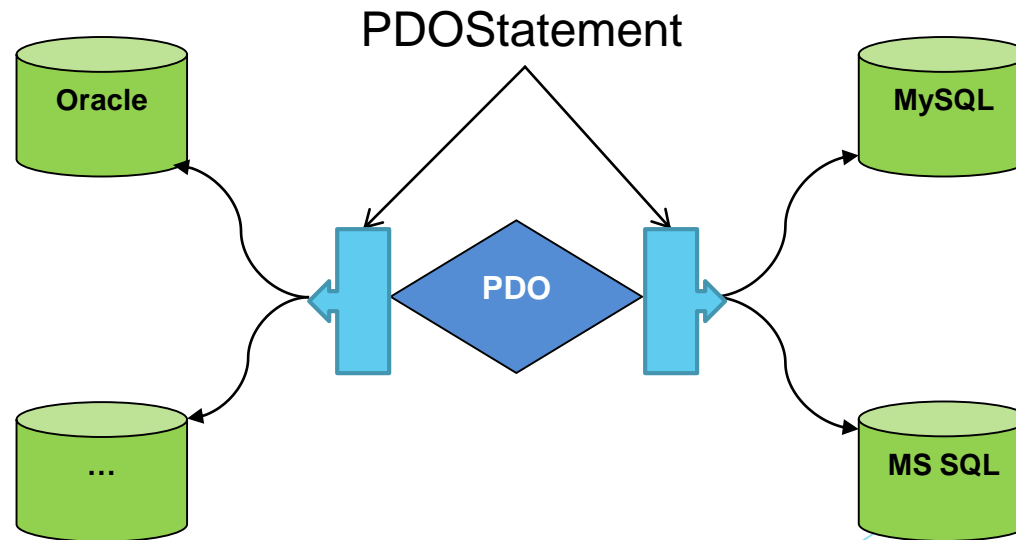


Mô hình làm việc giữa PDO và các loại CSDL



# 1. Tổng quan về PDO > Vì sao chọn PDO?

- An toàn: với giải pháp PDOStatement chúng ta không còn lo sợ SQL injection nữa.



Mô hình làm việc giữa PDO và các loại CSDL

# 1. Tổng quan về PDO

- ▶ PDO là gì?
- ▶ Vì sao chọn PDO?
- ▶ Tạo kết nối – đóng kết nối
- ▶ Thực hiện lệnh SQL

# 1. Tổng quan về PDO > Tạo kết nối

## ▶ Cú pháp:

```
$dbh = new PDO(<driver>, <username>, <password>,  
               <driver_options>);
```

## ▶ Tham số:

- ▶ <driver>: chuỗi tên driver dùng để làm việc CSDL.
- ▶ <username>: tên đăng nhập vào CSDL.
- ▶ <password>: mật khẩu dùng để truy cập vào CSDL.
- ▶ <driver\_options>: tham số tùy chọn, dùng để bổ sung thêm các trạng thái kết nối của PDO.

# 1. Tổng quan về PDO > Tạo kết nối

- ▶ Xem server hỗ trợ những loại driver nào:

```
<?php  
    print_r(PDO::getAvailableDrivers());  
?>
```

# 1. Tổng quan về PDO > Tạo kết nối

## ▶ Một số driver thông dụng:

- ▶ **PDO\_DBLIB ( FreeTDS / Microsoft SQL Server / Sybase )**
- ▶ PDO\_FIREBIRD ( Firebird/Interbase 6 )
- ▶ PDO\_IBM ( IBM DB2 )
- ▶ PDO\_INFORMIX ( IBM Informix Dynamic Server )
- ▶ **PDO\_MYSQL ( MySQL 3.x/4.x/5.x )**
- ▶ **PDO\_OCI ( Oracle Call Interface )**
- ▶ PDO\_ODBC ( ODBC v3 (IBM DB2, unixODBC and win32 ODBC) )
- ▶ PDO\_PGSQL ( PostgreSQL )
- ▶ PDO\_SQLITE ( SQLite 3 and SQLite 2 )
- ▶ PDO\_4D ( 4D )

# 1. Tổng quan về PDO > Tạo kết nối

- ▶ Ví dụ: Kết nối đến MySQL

```
$dbh = new PDO('mysql:host=localhost;dbname=test', 'root', '');
```

- ▶ Ví dụ: Kết nối đến MSSQL

```
$dbh = new PDO('dblib:host=localhost;dbname=test', 'root', '');
```

- ▶ Ví dụ: Kết nối đến Oracle

```
$tns = "(DESCRIPTION =  
    (ADDRESS_LIST =  
        (ADDRESS = (PROTOCOL = TCP)(HOST = yourip)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
        (SERVICE_NAME = orcl)  
    )  
)";
```

```
$dbh= new PDO("oci:dbname=". $tns, 'root', '');
```

# 1. Tổng quan về PDO > Hủy kết nối

- ▶ Cú pháp:  
**\$dbh = NULL;**

# 1. Tổng quan về PDO

- ▶ PDO là gì?
- ▶ Vì sao chọn PDO?
- ▶ Tạo kết nối – đóng kết nối
- ▶ Thực hiện lệnh SQL



# 1. Tổng quan về PDO > Thực hiện lệnh SQL

## Câu lệnh Insert,update,delete

- Cú pháp:

- `$dbh->exec(<query>);`

- Tham số:

- query: tham số bắt buộc, là câu lệnh truy vấn được gửi đi.

- `$sth->exec()`: Kết quả là tổng số dòng bị ảnh hưởng. Nếu không có dòng nào bị tác động thì kết quả là 0. Nếu không thực thi được câu lệnh thì kết quả là False

# 1. Tổng quan về PDO > Thực hiện lệnh SQL

Câu lệnh Insert,update,delete

- ▶ Ví dụ: Thực hiện lệnh xóa tất cả các dòng dữ liệu trong bảng hang\_sua

```
$dbh = new PDO('mysql:host=localhost;dbname=test', 'root', '');  
$sql= "DELETE FROM hang_sua";  
$count = $dbh->exec($sql);
```

# 1. Tổng quan về PDO > Thực hiện lệnh SQL

## Câu lệnh Select

- Cú pháp:

- \$dbh->query(<query>);

- Tham số:

- query: tham số bắt buộc, là câu lệnh truy vấn được gửi đi.

# 1. Tổng quan về PDO > Thực hiện lệnh SQL

## Câu lệnh Select

► Ví dụ: Thực hiện lệnh lấy thông tin loại sữa

```
$dbh = new PDO('mysql:host=localhost;dbname=test', 'root', '');  
$dbh->query("set names utf8");  
$arr = $dbh->query('select ma_loai_sua,ten_loai from loai_sua');  
foreach($arr as $row){  
    print_r($row);  
    echo "<br>"; }  
  
Array ( [ma_loai_sua] => SB [0] => SB [ten_loai] => Sữa bột [1] => Sữa bột )  
Array ( [ma_loai_sua] => SC [0] => SC [ten_loai] => Sữa chua [1] => Sữa chua )  
Array ( [ma_loai_sua] => SD [0] => SD [ten_loai] => Sữa đặc [1] => Sữa đặc )  
Array ( [ma_loai_sua] => ST [0] => ST [ten_loai] => Sữa tươi [1] => Sữa tươi )
```

# Nội dung

1. Tổng quan về PDO
2. Transaction
3. PDO Statement
4. Hiển thị dữ liệu
5. Thêm – xóa – cập nhật dữ liệu

## 2.Transaction

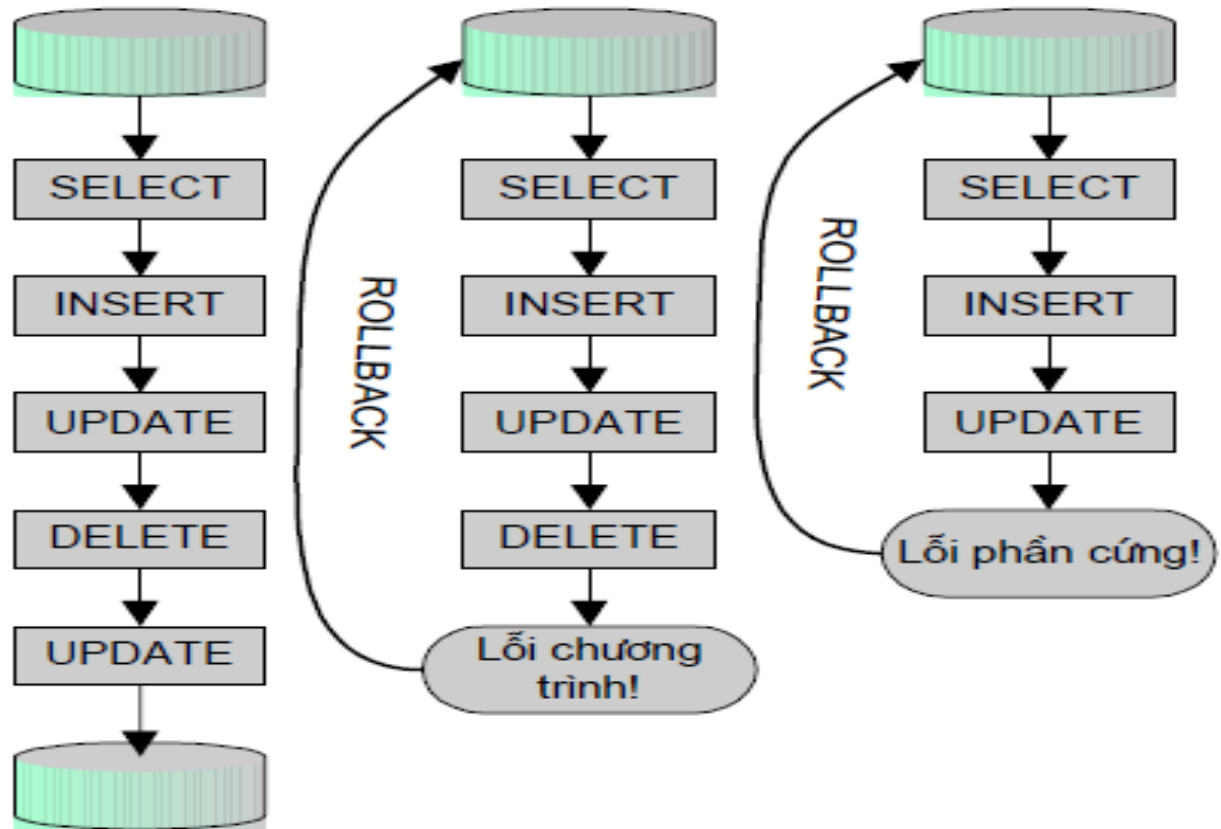
- ▶ Một giao tác (transaction) là một chuỗi một hoặc nhiều câu lệnh SQL được kết hợp lại với nhau thành một khối công việc. Việc kết hợp các câu lệnh lại với nhau trong một giao tác nhằm đảm bảo tính toàn vẹn dữ liệu và khả năng phục hồi dữ liệu.
- ▶ Trong một giao tác, các câu lệnh có thể độc lập với nhau nhưng tất cả các câu lệnh trong một giao tác đòi hỏi phải thực thi trọn vẹn hoặc không một câu lệnh nào được thực thi.

## 2.Transaction

Trạng thái CSDL  
trước khi giao tác  
tiến hành

Giao tác

Trạng thái CSDL  
sau khi giao tác  
tiến hành



*Hình 6.1: Giao tác SQL*

## 2.Transaction

- ▶ Giao tác SQL được định nghĩa dựa trên các câu lệnh xử lý giao tác sau đây:
  - ▶ BEGIN TRANSACTION: Bắt đầu một giao tác
  - ▶ COMMIT: Đánh dấu kết thúc giao tác. Câu lệnh này báo hiệu sự kết thúc thành công của một giao tác. Sau câu lệnh này, một giao tác mới sẽ được bắt đầu.
  - ▶ ROLLBACK: Hủy bỏ một giao tác và đưa cơ sở dữ liệu về trạng thái như trước khi giao tác bắt đầu. Một giao tác mới sẽ bắt đầu sau khi câu lệnh ROLLBACK được thực thi.



## 2.Transaction

### ► Ví dụ

```
$dbh->beginTransaction();  
$sth1 = $dbh->exec("delete from ct_hoadon where  
    so_hoa_don ='D002'");  
$sth2 = $dbh->exec("delete from hoa_don where  
    so_hoa_don ='D002'");  
if(!$sth1 || !$sth2)  
    $dbh->rollBack();  
else  
    $dbh->commit();
```

# Nội dung

1. Tổng quan về PDO
2. Transaction
3. PDO Statement
4. Hiển thị dữ liệu
5. Thêm – xóa – cập nhật dữ liệu

### 3. PDO statement

- ▶ Đếm số lượng mẫu tin
- ▶ Duyệt dữ liệu
- ▶ Truyền tham số trong câu truy vấn

### 3. PDO Statement > Đếm số lượng mẫu tin

- ▶ Cú pháp:

`PDOStatement->rowCount()`

- ▶ Hàm này có kết quả trả về là số lượng mẫu tin bị tác động bởi câu lệnh **insert, update, delete**

### 3. PDO Statement > Đếm số lượng mẫu tin

- ▶ **Ví dụ**: đếm số lượng mẫu tin bị xóa trong bảng sua có ma\_hang\_sua="AB"

```
$sql= "DELETE FROM sua WHERE ma_sua='AB' ";  
$sth = $dbh->prepare($sql);  
$sth->execute();  
echo $sth->rowCount(); // 9
```

### 3. PDO statement > Duyệt dữ liệu

- ▶ Duyệt lần lượt từng mẫu tin
- ▶ Duyệt danh sách các mẫu tin

### 3. PDO statement > Duyệt dữ liệu

Duyệt lần lượt từng mẫu tin

► Cú pháp:

```
PDOStatement->fetch([int $fetch_style] )
```

► \$fetch\_style:

PDO::FETCH\_NUM: kết quả trả về một mảng (có chỉ số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng \$row[0], \$row[1], \$row[2],...

### 3. PDO statement > Duyệt dữ liệu

#### Duyệt lần lượt từng mẫu tin

- ▶ PDO::FETCH\_ASSOC: kết quả trả về là một mảng (có chỉ số chuỗi) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng: `$row["tên cột 1"]`, `$row["tên cột 2"]`, `$row["tên cột 3"]`,...
- ▶ PDO::FETCH\_BOTH: kết quả trả về là một mảng (có chỉ số chuỗi và số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột → truy cập bằng cách gọi từng phần tử của mảng: `$row["tên cột 1"]`, `$row["tên cột 2"]`, `$row["tên cột 3"]`,... hoặc `$row[0]`, `$row[1]`, `$row[2]`,...



### 3. PDO statement > Duyệt dữ liệu

#### Duyệt lần lượt từng mẫu tin

- ▶ `PDO::FETCH_OBJ` :kết quả trả về là một mẫu tin trong bộ các mẫu tin như là một đối tượng → truy cập bằng cách gọi từng thuộc tính của đối tượng: `$tên_đối_tượng → tên_cột_1`, `$tên_đối_tượng → tên_cột_2,...`
- ▶ `PDO::FETCH_UNIQUE` :kết quả trả về là một mảng 2 chiều chứa danh sách kết quả đọc lên và chỉ lấy phần tử đầu tiên có cùng giá trị của cột đầu tiên trong câu lệnh sql. Và giá trị của cột đầu tiên sẽ làm khóa cho mảng con.
- ▶ `PDO::FETCH_GROUP` :kết quả trả về là một mảng 2 chiều chứa danh sách kết quả đọc lên theo từng nhóm với giá trị của cột đầu tiên làm khóa cho mảng con.

### 3. PDO statement > Duyệt dữ liệu

Duyệt danh sách các mẫu tin

▶ Cú pháp:

```
PDOStatement->fetchAll([int $fetch_style,[$column_index]])
```

### 3. PDO statement > Duyệt dữ liệu

#### Duyệt danh sách các mẫu tin

- ▶ Tham số:
  - ▶ **\$fetch\_style:**
    - ▶ *PDO::FETCH\_BOTH* : giá trị mặc định. Kết quả trả về là mảng hai chiều, mỗi phần tử là một mảng 1 chiều (có chỉ số chuỗi và số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột.
    - ▶ *PDO::FETCH\_COLUMN* : Kết quả trả về là mảng 1 chiều chứa tất cả các giá trị của một cột được chỉ định
  - ▶ **\$column\_index:** kết quả trả về là cột được chỉ định khi \$fetch\_style có giá trị là *PDO::FETCH\_COLUMN*

### 3. PDO statement > Duyệt dữ liệu

Duyệt danh sách các mẫu tin

Ví dụ: in giá trị của cột đầu tiên trong bảng khách hàng

```
...  
$sth = $dbh->prepare("select * from khách_hang");  
$sth->execute();  
$result = $sth->fetchAll(PDO::FETCH_COLUMN, 0);  
print_r($result);  
...
```

### 3. PDO Statement > Truyền tham số trong câu truy vấn

#### •Cú pháp

`PDOStatement->bindParam(mixed $parameter, mixed &$variable [, int $data_type [, int $length [, mixed $driver_options]]] )`

#### •Tham số:

- `$parameter`: tên tham số
- `$variable`: giá trị tham số
- `$data_type`: tùy chọn, kiểu dữ liệu của tham số, có cú pháp `PDO::PARAM_*`
- `$length`: tùy chọn, chiều dài của kiểu dữ liệu
- `$driver_options`

### 3. PDO statement > Truyền tham số trong câu truy vấn

```
$hang_sua = "AB";  
$don_gia = 100000;  
$sth = $dbh->prepare("select ma_sua,ten_sua FROM sua WHERE  
ma_hang_sua=:hang AND don_gia < :don_gia");  
$sth->bindParam(':don_gia',$don_gia);  
$sth->bindParam(':hang',$hang_sua);  
$sth->execute();  
$table = $sth->fetchAll(PDO::FETCH_ASSOC);  
print_r($table);
```

### 3. PDO statement > Truyền tham số trong câu truy vấn

- Ví dụ: liệt kê danh sách sữa có ma\_hang\_sua là AB và đơn giá nhỏ hơn 100000

```
$hang_sua = "AB";  
$don_gia = 100000;  
$sth = $dbh->prepare("select ma_sua,ten_sua FROM sua WHERE  
ma_hang_sua=? AND don_gia < ?");  
$sth->bindParam(1,$hang_sua);  
$sth->bindParam(2,$don_gia);  
$sth->execute();  
$table = $sth->fetchAll(PDO::FETCH_ASSOC);  
print_r($table);
```

# Nội dung

1. Tổng quan về PDO
2. Transaction
3. PDO Statement
4. Hiển thị dữ liệu
5. Thêm – xóa – cập nhật dữ liệu



## 4. Hiện thị dữ liệu

- ▶ Hiện thị dữ liệu không định dạng
  - ▶ Cách đơn giản nhất là hiện thị dữ liệu theo dạng bảng hoặc dạng text
  - ▶ Ví dụ: Hiện thị thông tin bảng khách hàng

THÔNG TIN KHÁCH HÀNG

| Mã KH | Tên khách hàng    | Giới tính | Địa chỉ                          | Số điện thoại |
|-------|-------------------|-----------|----------------------------------|---------------|
| kh001 | Khuất Thùy Phương | 1         | A21 Nguyễn Oanh quận Gò Vấp      | 9874125       |
| kh002 | Đỗ Lâm Thiên      | 0         | 357 Lê Hồng Phong Q.10           | 8351056       |
| kh003 | Phạm Thị Nhung    | 1         | 56 Đinh Tiên Hoàng quận 1        | 9745698       |
| kh004 | Nguyễn Khắc Thiện | 0         | 12bis Đường 3-2 quận 10          | 8769128       |
| kh005 | Tô Trần Hồ Giảng  | 0         | 75 Nguyễn Kiệm quận Gò Vấp       | 5792564       |
| kh006 | Nguyễn Kiến Thi   | 1         | 357 Lê Hồng Phong Q.10           | 9874125       |
| kh008 | Nguyễn Anh Tuấn   | 0         | 1/2bis Nơ Trang Long Q.BT TP.HCM | 8753159       |

# 4. Hiện thị dữ liệu

- ▶ Hiện thị dữ liệu có yêu cầu
  - ▶ Định dạng hiện thị dữ liệu
  - ▶ Tạo các cột tùy biến
  - ▶ Phân trang
  - ▶ Liên kết trang có chuỗi tham số

# 4. Hiện thị dữ liệu

- ▶ Định dạng hiển thị dữ liệu
  - ▶ Yêu cầu đơn giản nhất của một trang Web là dữ liệu hiển thị phải được định dạng giúp cho người dùng dễ dàng đọc thông tin.
  - ▶ Các định dạng thông thường là định dạng cột, dòng, kích thước, màu sắc, kiểu chữ,... cho nội dung hiển thị.
  - ▶ Để thực hiện được các yêu cầu về định dạng → cần phải xem xét và tính toán cho các nội dung cần hiển thị.

# 4. Hiện thị dữ liệu

- ▶ Định dạng hiển thị dữ liệu
  - ▶ Ví dụ: Hiện thị thông tin khách hàng với định dạng như sau:
    - ▶ Dòng tiêu đề của bảng sẽ được canh giữa, chữ in đậm, màu sắc phân biệt so với các dòng khác.
    - ▶ Các dòng thông tin của khách hàng: dòng chẵn và dòng lẻ có màu nền khác nhau, cột giới tính được canh giữa.

## 4. Hiện thị dữ liệu

► Định dạng hiện thị dữ liệu

► Ví dụ:

Dòng tiêu đề

### THÔNG TIN KHÁCH HÀNG

| Mã KH | Tên khách hàng    | Giới tính | Địa chỉ                          | Số điện thoại |
|-------|-------------------|-----------|----------------------------------|---------------|
| kh001 | Khuất Thùy Phương | 1         | A21 Nguyễn Oanh quận Gò Vấp      | 9874125       |
| kh002 | Đỗ Lâm Thiên      | 0         | 357 Lê Hồng Phong Q.10           | 8351056       |
| kh003 | Phạm Thị Nhung    | 1         | 56 Đinh Tiên Hoàng quận 1        | 9745698       |
| kh004 | Nguyễn Khắc Thiện | 0         | 12bis Đường 3-2 quận 10          | 8769128       |
| kh005 | Tô Trần Hồ Giảng  | 0         | 75 Nguyễn Kiêm quận Gò Vấp       | 5792564       |
| kh006 | Nguyễn Kiến Thi   | 1         | 357 Lê Hồng Phong Q.10           | 9874125       |
| kh008 | Nguyễn Anh Tuấn   | 0         | 1/2bis Nơ Trang Long Q.BT TP.HCM | 8753159       |

Dòng lẻ

45

Dòng chẵn

## 4. Hiện thị dữ liệu

- ▶ Tạo các cột tùy biến
  - ▶ Nhu cầu hiện thị dữ liệu một cách rõ ràng, chi tiết, dễ hiểu trở thành một yêu cầu tất yếu khi thiết kế trang Web.
  - ▶ Một trong những cách hiện thị dữ liệu đáp ứng nhu cầu đó là tạo các cột tùy biến

## 4. Hiện thị dữ liệu

- ▶ Tạo các cột tùy biến
  - ▶ Ví dụ: trong CSDL thường dùng kiểu dữ liệu boolean cho cột giới tính → cột này sẽ có một trong hai giá trị là 0 hoặc 1 và tùy theo quy ước để xác định 0 hoặc 1 thuộc giới tính gì. Nếu cột giới tính hiển thị 0, 1 và sau đó lại có dòng chú giải thì sẽ không thuận tiện → dùng hình ảnh hoặc checkbox,... để hiển thị giới tính. Cách hiển thị này vừa đẹp mắt, vừa dễ hiểu.

## 4. Hiện thị dữ liệu

### ► Tạo các cột tùy biến

#### ► Ví dụ:

### THÔNG TIN KHÁCH HÀNG

| Mã KH | Tên khách hàng    | Giới tính   | Địa chỉ                          | Số điện thoại |
|-------|-------------------|---|----------------------------------|---------------|
| kh001 | Khuất Thùy Phương |    | A21 Nguyễn Oanh quận Gò Vấp      | 9874125       |
| kh002 | Đỗ Lâm Thiên      |    | 357 Lê Hồng Phong Q.10           | 8351056       |
| kh003 | Phạm Thị Nhung    |    | 56 Đinh Tiên Hoàng quận 1        | 9745698       |
| kh004 | Nguyễn Khắc Thiện |   | 12bis Đường 3-2 quận 10          | 8769128       |
| kh005 | Tô Trần Hồ Giảng  |  | 75 Nguyễn Kiệm quận Gò Vấp       | 5792564       |
| kh006 | Nguyễn Kiến Thi   |  | 357 Lê Hồng Phong Q.10           | 9874125       |
| kh008 | Nguyễn Anh Tuấn   |  | 1/2bis Nơ Trang Long Q.BT TP.HCM | 8753159       |



## 4. Hiện thị dữ liệu

- ▶ Phân trang
  - ▶ Đối với những bảng có số lượng mẫu tin lớn khi hiện thị dữ liệu sẽ làm cho người dùng cảm thấy khó xem → thực hiện việc phân trang.
  - ▶ Phân trang dữ liệu giúp cho việc xem và tìm kiếm thông tin dễ dàng, nhanh chóng và thuận tiện.
  - ▶ Sử dụng class pager được cung cấp để thực hiện việc phân trang

# 4. Hiện thị dữ liệu

## ► Phân trang



## THÔNG TIN CHI TIẾT CÁC LOẠI SỮA

### Daisy Không Đường - Daisy



**Thành phần dinh dưỡng:**

**Lợi ích:**

- Chống loãng xương, giúp xương chắc khỏe.
- Phòng ngừa sự thiếu máu của trẻ, gia tăng hoạt động hệ tuần hoàn máu và não, giúp trẻ phát triển hoàn thiện.

**Trọng lượng: 900 gr - Đơn giá: 79.000 VNĐ**

### Daisy Vani - Daisy



**Thành phần dinh dưỡng:**

**Lợi ích:**

- Chống loãng xương, giúp xương chắc khỏe.
- Phòng ngừa sự thiếu máu của trẻ, gia tăng hoạt động hệ tuần hoàn máu và não, giúp trẻ phát triển hoàn thiện.
- Giúp hệ xương phát triển mạnh nhất.

**Trọng lượng: 454 gr - Đơn giá: 41.000 VNĐ**

<< < 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 > >>

## 4. Hiện thị dữ liệu

- ▶ Liên kết trang có chuỗi tham số

- ▶ Chuỗi URL có cấu trúc định dạng như sau:

`http://<host>[:<port>][path][?chuỗi_tham_số]`

- ▶ Phần cuối của chuỗi URL là chuỗi tham số có cú pháp như sau:

`[?Tham_số_1=Giá_trị_1[&Tham_số_2=Giá_trị_2[&...]]]`

- ▶ Chú ý: trong trường hợp có nhiều tham số, các cặp [Tham\_số=Giá\_trị] phân cách nhau bằng dấu &

## 4. Hiện thị dữ liệu

- ▶ Liên kết trang có chuỗi tham số
  - ▶ Ví dụ: Khi người dùng muốn xem thông tin chi tiết của sữa, người dùng click chọn sữa đó, URL sẽ có đường dẫn là địa chỉ của trang chi tiết sữa và chuỗi tham số là mã sữa



Sữa đặc có đường Ông Thọ Trắng

360

11.000

Trang danh sách sữa

[http://localhost/bai\\_tap\\_ph3/chi\\_tiet\\_sua.php?ma\\_sua=VNM001](http://localhost/bai_tap_ph3/chi_tiet_sua.php?ma_sua=VNM001)

Trang chi tiết sữa

### Sữa đặc có đường Ông Thọ Trắng - Vinamilk



#### Thành phần dinh dưỡng:

Được chế biến từ sữa bò cao cấp, bột sữa, chất béo, đường kính và vitamin A,D,B1 theo tiêu chuẩn quốc tế về giá trị và vệ sinh thực phẩm

#### Lợi ích:

Uống sữa Ông Thọ mỗi ngày sẽ giúp chống suy dinh dưỡng, giải độc, tăng tuổi thọ, phát triển chiều cao, tăng cường sức đề kháng.

**Trọng lượng: 360 gr - Đơn giá: 11.000 VNĐ**

## 4. Hiện thị dữ liệu

- ▶ Liên kết trang có chuỗi tham số

- ▶ Nhận giá trị truyền qua tham số:

`$_GET["tên_tham_số"]`

hoặc `$_REQUEST["tên_tham_số"]`

- ▶ Ví dụ: nhận giá trị của tham số mã sữa

```
$ma_sua = $_GET["ma_sua"];
```

# Nội dung

1. Tổng quan về PDO
2. Transaction
3. PDO Statement
4. Hiển thị dữ liệu
5. Thêm – xóa – cập nhật dữ liệu

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ▶ Thêm mới

- ▶ Sử dụng phương thức `exec()`/ `execute()` kết hợp với câu lệnh truy vấn `INSERT INTO`
- ▶ Kết quả trả về của `exec()`/ `execute()` là tổng số dòng bị ảnh hưởng. Nếu không có dòng nào thì kết quả là **0**. Nếu không thực thi được câu lệnh thì kết quả là **False**

- ▶ trong đó, `<query>`: tham số bắt buộc, là câu lệnh truy vấn được gửi đi

```
$db->exec(<query>);
```

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ► Thêm mới

- Ví dụ: Thêm mới một hãng sữa với giá trị cụ thể vào bảng hãng sữa đã có (dùng exec()).

```
<?php
$db = new
PDO('mysql:host=localhost;dbname=ql_ban_sua;charset=utf8'
, 'root', '123456');
$query = "INSERT INTO hang_sua VALUES ('XO', 'XO', 'Khu
công nghiệp Đồng Nai', '7456321', 'xo@xo.com')";
$num_rows = $db->exec($query);
echo $num_rows.' dòng đã thêm mới';
?>
```



# 5. Thêm mới – xóa – cập nhật dữ liệu

## ► Thêm mới

- Ví dụ: Thêm mới một hãng sữa với thông tin do người dùng nhập vào trên Form (execute())

```
<?php
$db = new PDO('mysql:host=localhost;dbname=ql_ban_sua;charset=utf8', 'root',
'123456');
// lấy thông tin của hãng sữa do người dùng nhập, kiểm tra thông tin, sau đó
gán vào các biến: $ma_hang_sua, $ten_hang_sua, $dia_chi, $dien_thoai, $email
...
// thêm hãng sữa mới vào bảng hang_sua
$sth = $db->prepare("INSERT INTO hang_sua VALUES (?, ?, ?, ?, ?)");
$sth->bindParam(1, $ma_hang_sua, PDO::PARAM_STR);
$sth->bindParam(2, $ten_hang_sua, PDO::PARAM_STR);
$sth->bindParam(3, $dia_chi, PDO::PARAM_STR);
$sth->bindParam(4, $dien_thoai, PDO::PARAM_STR);
$sth->bindParam(5, $email, PDO::PARAM_STR);
$sth->execute();
?>
```

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ▶ Xóa dữ liệu

- ▶ Sử dụng phương thức `exec()`/ `execute()` kết hợp với câu lệnh truy vấn `DELETE`
  - ▶ **Chú ý:** Cú pháp `DELETE` dùng để xóa một hay nhiều mẫu tin đã có trong bảng của CSDL đã được hướng dẫn ở bài trước

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ► Xóa dữ liệu

### ► Ví dụ: Xóa thông tin của hãng sữa XO

```
<?php
$db = new
PDO('mysql:host=localhost;dbname=ql_ban_sua;charset=utf8'
, 'root', '123456');

$ma_hang_sua = 'KO';
$stmt = $db->prepare("DELETE FROM hang_sua WHERE
Ma_hang_sua = ?");
$stmt->bindParam(1, $ma_hang_sua, PDO::PARAM_STR);

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

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ▶ Cập nhật dữ liệu

- ▶ Sử dụng phương thức `exec()`/ `execute()` kết hợp với câu lệnh truy vấn `UPDATE`.
  - ▶ Cú pháp `UPDATE` dùng để cập nhật nội dung của một mẫu tin đã có trong một bảng của CSDL đã được hướng dẫn ở bài trước.

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ▶ Cập nhật dữ liệu

- ▶ Sử dụng phương thức `exec()`/ `execute()` kết hợp với câu lệnh truy vấn `UPDATE`.
  - ▶ Cú pháp `UPDATE` dùng để cập nhật nội dung của một mẫu tin đã có trong một bảng của CSDL đã được hướng dẫn ở bài trước.

# 5. Thêm mới – xóa – cập nhật dữ liệu

## ► Cập nhật dữ liệu

- Ví dụ: Cập nhật lại địa chỉ cho hãng sữa XO là 'Khu công nghiệp Bình Dương'

```
<?php
$db = new PDO('mysql:host=localhost;dbname=ql_ban_sua;charset=utf8',
    'root', '123456');
$ma_hang_sua = 'ME';
$dia_chi = 'Khu CN Sóng Thần - Bình Dương';
$stmt = $db->prepare("UPDATE hang_sua SET Dia_chi = ? WHERE Ma_hang_sua = ?");
$stmt->bindParam(1, $dia_chi, PDO::PARAM_STR);
$stmt->bindParam(2, $ma_hang_sua, PDO::PARAM_STR);

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

# Thảo luận

