

Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh

Khoa Công nghệ Thông tin

Trường Đại học Công nghệ, ĐHQGHN

E-mail: thanhd@vnu.edu.vn Mobile: 0987.257.504

Chương 7

Thao tác CSDL

Nội dung

- Tổng quan
- DO
- ORM

Tổng quan

- Các bước chính

1. Kết nối đến máy chủ CSDL và lựa chọn CSDL để làm việc
2. Xây dựng các truy vấn, cập nhật và thực hiện truy vấn, cập nhật
3. Xử lý dữ liệu kết quả trả về khi thực hiện các truy vấn, cập nhật
4. Đóng kết nối đến máy chủ cơ sở dữ liệu

- Hệ quản trị CSDL

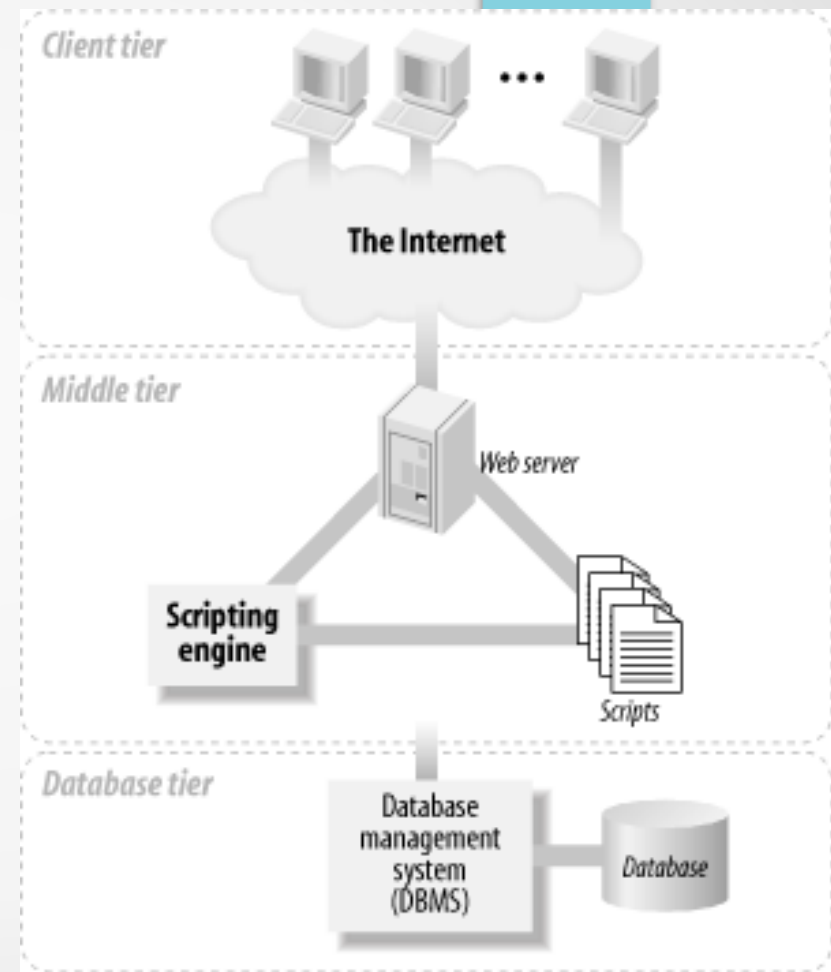
- MySQL, PostgreSQL, Oracle, MS SQL, MongoDB ...

- Thư viện thao tác CSDL

- DO
- ORM

Mô hình ba tầng

- **Tầng khách:** trình diễn và tương tác với người dùng
- **Tầng giữa:** thực hiện các logic của ứng dụng
- **Tầng CSDL:** bao gồm hệ quản trị CSDL, CSDL của ứng dụng



Tầng CSDL

- CSDL cho ứng dụng
- Hệ quản trị CSDL quản lý CSDL của ứng dụng
 - Nhận truy vấn từ các chương trình dịch tầng giữa, thao tác CSDL và trả kết quả cho chương trình yêu cầu

Sử dụng PDO

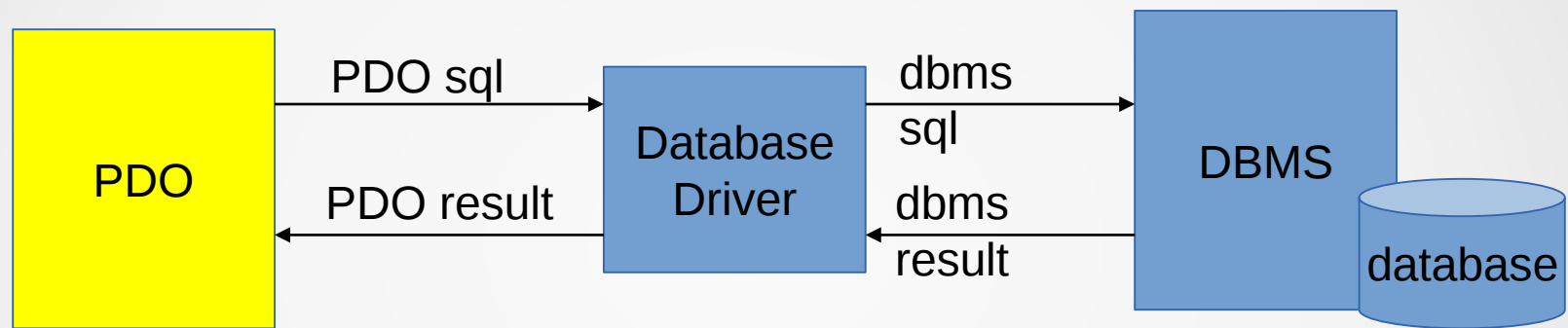
- từ PHP 5 trở đi
- hướng đối tượng
- sử dụng với mọi hệ QTCSDL
- nên dùng nếu viết mã mới

(Quá khứ: mysql, mysqli)

PHP Data Objects

- PDO cung cấp một giao diện chuẩn hóa, nhất quán để thao tác với CSDL sử dụng PHP
 - Dễ chuyển đổi CSDL
 - Viết mã hiệu quả và an toàn

Nguyên lý hoạt động



Cần cài driver cho hệ QTCSDL được sử dụng

Cài đặt và cấu hình

- PDO
 - Cài đặt *php5-mysql*,
 - Cấu hình *extension=pdo.so* trong *php.ini*
- Trình điều khiển cho MySQL
 - Cài đặt *pdo-mysql*
 - Cấu hình *extension=pdo_mysql.so* trong *php.ini*

Kết nối và đóng kết nối CSDL

- `$db = new PDO($dsn, $username, $password, [$options]);`
- Tên nguồn dữ liệu (dsn - database source name) khác nhau với các hệ QTCSDL
 - MySQL: `"mysql:host=...; dbname=...;"`
 - PostgreSQL: `"pgsql:host=...; dbname=...;"`
 - MS SQL: `"sqlsrv:Server=...;Database=..."`
 - ...
- Đóng kết nối
 - `$db = null;`

Xử lý ngoại lệ

```
try {  
    //Sử dụng PDO  
} catch(PDOException $ex) {  
    echo "Error".$ex->getMessage();  
}
```

Kết nối và đóng kết nối CSDL

```
try {  
    //Mở kết nối cơ sở dữ liệu  
    $dsn = "mysql:host=112.137.129.123;  
    dbname=thuvien;";  
    $db = new PDO($dsn, "canbo", "canbo@123");  
    //Các thao tác cơ sở dữ liệu: truy vấn, cập nhật  
    //  
    //Đóng kết nối cơ sở dữ liệu  
    $db = null;  
} catch (PDOException $e) {  
    echo "Lỗi".$e->getMessage();  
}
```

Cập nhật CSDL

```
try {  
    //Nhận dữ liệu từ request  
    $t = $_POST["tieude"];  
    $id = $_POST["id"];  
    //Cập nhật cơ sở dữ liệu  
    $r = $db->exec("update Sach set tieude='$t'  
    where id='$id';");  
    echo "$r bản ghi đã được cập nhật.";  
} catch (PDOException $e) {  
    echo "Lỗi".$e->getMessage();  
}
```

Cập nhật CSDL (tiếp)

```
try {  
    //Nhận dữ liệu từ request  
    $t = $_POST["tieude"];  
    $id = $_POST["id"];  
    //Chuẩn bị câu lệnh cập nhật  
    $stmt= $db->prepare("update Sach set tieude = ?  
    where id = :id");  
    //Buộc các tham số  
    $stmt->bindValue(1, $t);  
    $stmt->bindValue(":id", $id);  
    //Thực thi  
    $stmt->execute();  
} catch (PDOException $e) {  
    echo "Lỗi".$e->getMessage();  
}
```

1

Truy vấn CSDL

```
try {  
    //Truy vấn và xử lý kết quả  
    $stmt = $db->query("select * from Sach where  
    nxb='$nxb';");  
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {  
        echo $row['tieude']." ".$row['nxb']."\n";  
    }  
} catch (PDOException $e) {  
    echo "Lỗi".$e->getMessage();  
}
```

Truy vấn CSDL (tiếp)

```
try {  
    //Chuẩn bị câu lệnh truy vấn  
    $stmt= $db->prepare("select * from Sach where nxb = ?");  
    //Buộc các tham số  
    $stmt->bindValue(1, $nxb);  
    //Thực thi  
    $stmt->execute();  
    //Xử lý kết quả  
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {  
        echo $row['tieude']." ".$row['nxb']."\n";  
    }  
} catch (PDOException $e) {  
    echo "Lỗi".$e->getMessage();  
}
```

Xử lý an ninh dữ liệu nhập

- Dữ liệu do người dùng nhập có thể không đúng định dạng và chứa các ký tự đặc biệt làm cho chương trình không làm việc hoặc làm việc không đúng
 - Cần xử lý hợp thức dữ liệu trước khi sử dụng
- ...

Xử lý an ninh cho dữ liệu nhập

- Giới hạn độ dài của giá trị
- Thay các ký tự đặc biệt như ; . > < \ “ ‘ ... bằng cách thêm \ trước ký tự
- Ví dụ
 - Dữ liệu được nhập: 2001;cat /etc/passwd
 - Cắt bớt và thay thế dấu đặc biệt: 2001\;cat
- PHP

```
function clean($input, $maxlength=0) {  
    if ($maxlength > 0) $input = substr($input, 0, $maxlength);  
    if (function_exists('mysql_real_escape_string')) {  
        if (get_magic_quotes_gpc()) {  
            // Undo magic quote effects  
            $input = stripslashes($input);  
        }  
        // Redo escape using mysql_real_escape_string  
        $input = mysql_real_escape_string($input);  
    } else {  
        if (!$this->get_magic_quotes_gpc()) {  
            // Add slashed manually  
            $value = addslashes($input);  
        }  
    }  
    return $input;  
}
```

Giao tác

```
try {  
    $db->beginTransaction();  
    $db->exec("SOME QUERY");  
    $stmt = $db->prepare("SOME OTHER QUERY?");  
    $stmt->execute(array($value));  
    $stmt = $db->prepare("YET ANOTHER QUERY??");  
    $stmt->execute(array($value2, $value3));  
    $db->commit();  
} catch(PDOException $ex) {  
    $db->rollBack();  
    echo $ex->getMessage();  
}
```

Truy cập đồng thời

- Nhiều người truy cập CSDL đồng thời có thể dẫn đến các tình huống sau:
 - *Mất cập nhật*: Người dùng A đọc giá trị từ CSDL. Người dùng B cũng đọc giá trị từ CSDL và cập nhật ngay lập tức. Người dùng A cập nhật, ghi đè giá trị của người B
 - *Đọc sai*: Người dùng A cập nhật giá trị. Người dùng B đọc giá trị đã cập nhật. Người dùng A undo lại thao tác => Giá trị B đọc được không còn đúng
 - *Tính tổng sai*: Người dùng A đang tính tổng thì người dùng B thay đổi giá trị một số mục
 - *Đọc giá trị không thể lặp*: A đọc giá trị, B thay đổi giá trị, A đọc lại thấy giá trị khác

Truy cập đồng thời (tiếp)

- Khóa bảng cần thao tác
 - Kiểu khóa READ: cho người dùng khác được đọc nhưng không được ghi
 - Kiểu khóa WRITE: không cho người dùng khác đọc hay ghi
- Thực hiện truy vấn
- Thực hiện cập nhật
- Mở khóa bảng

Ví dụ Xử lý cập nhật đồng thời

//Khóa các bảng cần thao tác

```
$query = "LOCK TABLES items READ, orders WRITE, customer READ";
```

```
if (!$db->exec($query)) showerror();
```

// Thực hiện truy vấn

```
$query = "SELECT SUM(price*qty) from FROM items...";
```

```
if (!$stmt= $db->query($query)) showerror( );
```

...

//rồi cập nhật

```
if ($row["SUM(price*qty)"] > $minimum) {
```

```
    $query = "UPDATE orders SET discount = $discount WHERE cust_id =  
    $custId AND order_id = $orderId";
```

```
    if (!$db->exec($query)) showerror();
```

```
}
```

// Mở khóa các bảng

```
$db->exec("UNLOCK TABLES");
```

Lưu ý khi khóa các bảng

- Sau khi khóa xong phải mở khóa các bảng
- Không cần khóa các bảng nếu chỉ thực hiện một truy vấn.

ORM

ORM

- Mỗi bảng trong CSDL được ánh xạ vào một lớp (class)
- Mỗi bản ghi trở thành một đối tượng (object)

Tạo lớp ORM

```
<?php
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Book extends Model {  
    protected $table='Sach';  
}
```

Truy vấn dữ liệu

Sử dụng kết hợp các phương thức *all()*, *where()*, *lists()*, ...của model:

```
$books = Book::all();
```

```
$book = Book::where('id', '=', 1);
```

```
$book = Book::where('id', '=', 1) -> value('tieude');
```

```
$books = Book::lists('tieude');
```

```
$books = Book::where('tentacgia', '=', 'Nguyen Van A')  
    -> where('namxb', '<', '2017')->all();
```

```
$books = Book::where('tentacgia', '=', 'Nguyen Van A')  
    ->take(5)->get();
```

```
$books = Book::orderBy('namxb', 'desc')->get();
```

```
$c = Book::where('tentacgia', '=', 'Nguyen Van A')->count();
```

Thêm mới, cập nhật, xóa

```
<?php
```

```
use App\Book;  
$book = new Book();  
$book->id = '10';  
$book->tieude = 'Khởi nguồn sáng tạo';  
$book->save();
```

```
$book = Book::find(10);  
$book->tieude = 'Rèn luyện kỹ năng sáng tạo';  
$book->save();
```

```
$book = Book::find(10);  
$book->delete();
```

Quan hệ một–một

```
<?php
// Tập app/Models/Reader.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Reader extends Model {
    protected $table = 'Docgia';
    public function card() {
        return $this->hasOne('App\Models\Card', 'madg');
    }
}
```

```
<?php
// Tập app/Models/Card.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Card extends Model {
    protected $table = 'Thedoc';
    public function reader() {
        return $this->belongsTo('App\Models\Reader', 'madg');
    }
}
```

```
$card = Reader::find(100)->card;
```

=>

```
"select * from Thedoc where
madg=100"
```

```
$reader = Card::find(100)-
>reader;
```

=>

```
"select * from Docgia where
madg in (select madg from
Thedoc where mathe=100)"
```

Quan hệ một-nhiều

```
<?php
// Tập app/Models/Reader.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Reader extends Model
{
    protected $table = 'Docgia';
    public function books()
    {
        return $this->hasMany('App\Models\Book', 'madg');
    }
}

<?php
// Tập app/Models/Book.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Book extends Model {
    protected $table = 'Sach';
    public function reader() {
        return $this->belongsTo('App\Models\Reader', 'madg');
    }
}
```

```
$books = Reader::find(100)->books;
```

=>

```
"select * from Sach where madg=100"
```

```
$reader = Book::find(100)->reader;
```

=>

```
"select * from Docgia where madg in (select madg from Sach where masach=100)"
```

Quan hệ nhiều–nhiều

```
<?php
// Tập app/Models/Author.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Author extends Model {
    protected $table = 'Tacgia';
    public function books() {
        return $this->belongsToMany('App\Models\Book', 'VietSach', 'matg', 'masach');
    }
}
```

```
<?php
// Tập app/Models/Book.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Book extends Model {
    protected $table = 'Sach';
    public function authors() {
        return $this->belongsToMany('App\Models\Author', 'VietSach', 'masach', 'matg');
    }
}
```

```
$books = Author::find(100)->books;
=>
```

```
"select * from Sach where masach in
(select masach from VietSach where
    matg=100)"
```

Một số thư viện ORM

- PHP: Eloquent
- Ruby: ActiveRecord
- Java: Hibernate
- Python: SQLAlchemy
- Rust: SeaORM
- .NET: Entity
- JS/TS: Sequelize

Tiếp theo
Lưu trạng thái và đảm bảo an ninh

