# Bài 6: Load Library Database Trong Codeigniter

Trong bài này bạn sẽ được học:

- Các thao tác xử lý database.
- Làm quen với Active Record.

## 1. Load library database

Đầu tiên, chúng ta cần tạo ra một controller có tên là **user** và action **index**, chúng ta sẽ gọi library database ngay tại constructor. Việc gọi library ngay tại constructor giúp chúng ta có thể tái sử dụng các library này trong các action sau đó.

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){

    }
}
```

Như thế này là chưa đủ để chúng ta có thể sử dụng library, các bạn cần phải khai báo database ở trong thư mục *application/config/database.php* (các bạn xem lại bài Load Model Trong Codeigniter). Tiếp theo, chúng ta cần có một table với tên là user (các bạn có thể đặt tùy ý).

```
CREATE TABLE `user` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(100) COLLATE utf8_unicode_ci DEFAULT NULL,
  `password` char(32) COLLATE utf8_unicode_ci DEFAULT NULL,
  `level` int(1) DEFAULT '1',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8
COLLATE=utf8_unicode_ci
```

## 2. Thao tác Database

Đầu tiên, chúng ta sẽ tìm hiểu cách viết câu truy vấn bình thường, sau đó mới thao tác tới các active record do CI cung cấp. Để viết được các câu truy vấn bình thường, chúng ta dùng thuộc tính như sau:

```
$this->db->query("SELECT * FROM USER order By id");
```

Để có thể lấy ra toàn bộ dữ liệu trong table user ta có thao tác như sau.

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
```

```
        $this->load->library("database");
    }

    public function index(){
        $query=$this->db->query("SELECT * FROM USER order By id");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Ở đây, chúng ta dùng biến $data để chứa kết quả trả về, hàm ***result_array*** tương ứng với hàm ***mysql_fetch_assoc*** của ***PDO***. Để xem kết quả trả về, chúng ta chạy link *localhost/codeig/index.php/user/index.*

```
[0] => Array
(
[id] => 1
[username] => admin
[password] => 123456
[level] => 2
)
[1] => Array
(
[id] => 2
[username] => alibaba
[password] => 123456
[level] => 1
)
[2] => Array
(
[id] => 3
[username] => vicky
[password] => 123456
[level] => 1
)
[3] => Array
(
[id] => 4
[username] => jacky
[password] => dsads
[level] => 1
)
```

Ok, nếu kết quả trên trình duyệt hiển thị như thế này, xem như việc lấy toàn bộ record của bạn đã thành công rồi đấy. Tiếp theo, chúng ta sẽ sử dụng **active record** trong quá trình làm việc với database.

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }
```

```
    public function index(){
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Khi dùng phương thức get(), chúng ta chỉ cần truyền vào tên bảng. Việc sử dụng phương thức này giúp chúng ta rút ngắn câu lệnh một cách khá là dễ dàng. Ngoài ra, chúng ta cũng thể cài đặt một số ràng buộc cho câu truy vấn thông qua một số phương thức sau:

```
$this->db->limit();
$this->db->select();
$this->db->join();
$this->db->like();
$this->db->select_min();
$this->db->select_max();
```

Để trả về số lượng dòng dữ liệu, chúng ta sử dụng phương thức `$query->num_rows`.

Để trả về chỉ một record đầu tiên, chúng ta dùng: `$query->row_array();`

Một số ví dụ như sau:

Chúng ta không muốn liệt kê tất cả các field mà chỉ liệt kê một số field nào đó, chúng ta dùng phương thức select:

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){
        $this->db->select("id, username, level");
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Để sắp xếp id theo chiều giảm dần, chúng ta dùng phương thức order_by như sau:

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){
```

```
        $this->db->select("id, username, level");
        $this->db->order_by("id", "desc");
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Tương tự để giới hạn kết quả trả về, chúng ta sử dụng phương thức limit() với đối số thứ nhất là tổng số record trả về và đối số thứ 2 chính là vị trí bắt đầu để lấy record. (Nó khác câu truy vấn bình thường ở chỗ hoán đổi vị trí 2 đối số này).

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){
        $this->db->select("id, username, level");
        $this->db->order_by("id", "desc");
        $this->db->limit(7, 0);
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Để thực thi mệnh đề điều kiện, ví dụ trả về các user có level = 2 thì chúng ta có cú pháp như sau:

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){
        $this->db->select("id, username, level");
        $this->db->order_by("id", "desc");
        $this->db->limit(7, 0);
        $this->db->where("level", "2");
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Tiếp theo, để trả về phần tử lớn nhất, phần tử nhỏ nhất, chúng ta sử dụng phương thức select_max và select_min như ví dụ sau:

```
class User extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library("database");
    }

    public function index(){
        $this->db->select("id, username, level");
        $this->db->order_by("id", "desc");
        $this->db->limit(7, 0);
        $this->db->where("level", "2");
        $this->db->select_min("id");
        $query=$this->db->get("user");
        $data=$query->result_array();
        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

Tiếp theo, chúng ta sẽ tìm hiểu ba thao tác quan trọng khác khi làm việc với CSDL, đó là các thao tác: thêm dữ liệu (insert), cập nhật dữ liệu (update), xóa dữ liệu (delete).

Thêm dữ liệu:

Cú pháp: `$this->db->insert("table", tên biến truyền vào)`

Ví dụ:

```
public function index2(){
        $data=array(
            "username" => "Huỳnh Tấn Khải",
            "password" => "1212445",
            "level"     => "2",
        );
        $this->db->insert("user", $data);
    }
```

Kiểm tra bằng cách, chạy link *localhost/codeig/index.php/user/index2* , sau đó quay trở lại action index xem kết quả.

Cập nhật dữ liệu:

Cú pháp: `$this->db->update("table", tên biến truyền vào)`

Ví dụ:

```
public function index3(){
        $data=array(
            "username" => "Huỳnh Tấn Khải",
            "password" => "123456",
            "level"     => "1",
        );
        $this->db->where("id", "5");
        if($this->db->update("user", $data)){
```

```
            echo "Update Thanh cong";
        }else{
            echo "Update That bai";
        }
    }
```

Chúng ta đã tiến hành update user có id = 5, nếu màn hình trình duyệt trả về là "Update Thanh Cong" thì việc cập nhật dữ liệu đã thành công. Kiểm tra bằng cách chạy link *localhost/codeig/index.php/user/index3*, quay trở lại action index để xem kết quả.

Cập nhật dữ liệu:

Cú pháp: `$this->db->delete("table")`

Ví dụ:

```
public function index4(){
        $this->db->where("id", "5");
        if($this->db->delete("user")){
            echo "Xoa thanh cong";
        }else{
            echo "Xoa that bai";
        }
    }
```

Ở đây chúng ta đã xóa người dùng có id = 5, chạy link *localhost/codeig/index.php/user/index4* để tiến hành xóa và quay lại action index để xem kết quả.

Như vậy chúng ta đã tìm hiểu các thao tác làm việc với CSDL trong CI. Các bạn chú ý rằng các thao tác này không nên viết trong controller vì nó trái với nguyên tắc của CI, bản thân của controller không tương tác trực tiếp với database, để làm việc tốt với CSDL, chúng ta nên viết nó trong model.

## 3. Thao tác database trong model

Đầu tiên, chúng ta cần phải khởi tạo file model và chúng ta cũng cần ghi nhớ là tên file của model không được phép trùng tên với controller, nếu 2 file này trùng tên nhau (dù ở 2 thư mục khác nhau) thì CI sẽ báo lỗi ngay, tại thư mục **application/models,** chúng ta khởi tạo file UserModel.php và khai báo như sau:

```
<?php
class UserModel extends CI_Model{
    public function __construct(){
        parent::__construct();
        $this->load->database();
    }
}
?>
```

Để tiến hành lấy toàn bộ record trong table user, chúng ta tạo một function có tên là listUser. Function này thao tác với bảng user và trả kết quả về thông qua câu lệnh return.

```
<?php
class UserModel extends CI_Model{
    public function __construct(){
        parent::__construct();
        $this->load->database();
    }

    public function listUser(){
        $this->db->select("id, username, level");
        $query=$this->db->get("user");
        return $query->result_array();
    }
}
?>
```

Tại controller user, chúng ta tạo action index5, action này sẽ gọi model UserModel để lấy dữ liệu và truyền vào view để hiển thị. Để hiển thị dữ liệu, chúng ta tạo view có tên là **list_view.php** (trong thư mục **application/views**) có nội dung như sau:

```
echo "<pre>";
print_r($data);
echo "</pre>";
```

Như vậy, nội dung của action index5 như sau:

```
public function index5(){
       $this->load->model("UserModel");
       $data['data']=$this->UserModel->listUser();
       $this->load->view("user/list_view", $data);
}
```

Các bạn truy cập link *localhost/codeig/index.php/user/index5* để kiểm tra.