

## Lab 6: Database: Eloquent



### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu được các thao tác giữa ứng dụng và database
- ✓ Cấu hình và sử dụng Eloquent
- ✓ Thao tác CRUD

### BÀI 1 (2 ĐIỂM)

Eloquent models kế thừa từ lớp `Illuminate\Database\Eloquent\Model`. Eloquent ORM (Object Relational Mapper) được Laravel phát triển giúp việc tương tác, làm việc với cơ sở dữ liệu trở nên dễ dàng hơn. Về nguyên tắc thì đặt tên table ở dạng số nhiều, ta sẽ có tên model ở dạng số ít, ví dụ tên table là « categories » thì tên model sẽ là « Category », đồng thời model được lưu trong `/app`.

### HƯỚNG DẪN:

- ✓ Tạo model tên `BaseModel`

```
1 | php artisan make:model BaseModel
```

- ✓ Sau khi tạo thành công, mở `BaseModel.php` trong `/app/BaseModel.php` để update code như sau

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6  use DB;
7
8  class BaseModel extends Model {
9
10     public function selectQuery($sql_stmt) {
11         return DB::select($sql_stmt);
12     }
13
14     public function sqlStatement($sql_stmt) {
15         DB::statement($sql_stmt);
16     }
17 }

```

- `namespace App;` Tạo namespace cho base model
- `use Illuminate\Database\Eloquent\Model;` imports để sử dụng Eloquent ORM model
- `use DB;` sử dụng DB namespace
- `public function selectQuery($sqlstmt) { return DB::select($sqlstmt); }` định nghĩa hàm `selectQuery($sql_stmt)`. `$sqlstmt` nhận vào tham số là câu lệnh SQL.  
`DB::select($sqlstmt)` và thực thi câu lệnh đó
- `public function sqlStatement($sqlstmt) { DB::statement($sqlstmt); }` Tạo hàm thực thi câu lệnh SQL được truyền vào. `$sqlstmt` `DB::statement($sqlstmt)`

✓ Sử dụng modle Category (sinh viên tạo thêm) để thêm record vào trong database, mở model /app/Category.php và thêm code

```

1  <?php
2
3  namespace App;
4
5  class Category extends BaseModel {
6      protected $primaryKey = 'id';
7      protected $table = 'categories';
8      protected $fillable = array('name', 'created_at_ip', 'updated_at_ip');
9  }

```

## FPT POLYTECHNIC

- `namespace App;` khai báo namespace
- `class Category extends BaseModel` Tạo class “category” kế thừa từ `BaseModel`
- `protected $primaryKey = 'id';` Xác định khóa chính
- `protected $table = 'categories';` Xác định tên table.
- `protected $fillable = array('name', 'createdatip', 'updatedatip');` Tạo các field

- ✓ Để thực hiện thêm dữ liệu vào cơ sở dữ liệu thông qua model category, sinh viên mở `/app/Http/routes.php` và thêm code

```
1 | Route::get('/insert', function() {
2 |     App\Category::create(array('name' => 'Music'));
3 |     return 'category added';
4 | });
```

- `App\Category::create(array('name' => 'Music'));` Mảng `('name' => 'Music')` cung cấp giá trị “music” cho field “names” trong table.
- `return 'category added';` hiển thị kết quả “category added” trên web browser

- ✓ chạy trình duyệt và kiểm tra kết quả

```
1 | http://localhost/larashop/public/insert
```

```
1 | category added
```

## BÀI 2 (3 ĐIỂM):

Dùng Eloquent đọc dữ liệu, sử dụng model category đọc tất cả dữ liệu từ cơ sở dữ liệu.

### HƯỚNG DẪN:

- ✓ Mở file `/app/Http/routes.php` và thêm code

```

1 Route::get('/read', function() {
2     $category = new App\Category();
3
4     $data = $category->all(array('name','id'));
5
6     foreach ($data as $list) {
7         echo $list->id . ' ' . $list->name . ' ';
8     }
9 });

```

- `$category = new App\Category();` Tạo một thể hiện của Category model
- `$data = $category->all(array('name','id'));` Gọi các phương thức có trong model. Mảng với các tham số ('name','id') dùng để xác định column names mà câu truy vấn trả kết quả về.
- `foreach ($data as $list){...}` Duyệt qua các kết quả trả về và hiển thị lên trình duyệt

✓ Kiểm tra kết quả khi chạy trên trình duyệt

```
1 | http://localhost/larashop/public/read
```

Tùy vào dữ liệu sinh viên nhập trong categories sẽ nhận được kết quả tương ứng

```

1 5 CLOTHING
2 4 FASHION
3 3 KIDS
4 1 MENS
5 16 Music
6 2 WOMENS

```

### BÀI 3 (3 ĐIỂM)

Update dữ liệu dựa vào Id, giả sử chỉnh sửa category “music” có id 16 thành HEAVY METAL. Mở /app/Http/routes.php và thêm code:

```

1 Route::get('/update', function() {
2     $category = App\Category::find(16);
3     $category->name = 'HEAVY METAL';
4     $category->save();
5
6     $data = $category->all(array('name', 'id'));
7
8     foreach ($data as $list) {
9         echo $list->id . ' ' . $list->name . ' ';
10    }
11 });

```

- `$category = App\Category::find(16);` Tìm record có id “16”
  - `$category->name = 'HEAVY METAL';` Gán giá trị HEAVY METAL vào field name
  - `$category->save();` xác nhận lưu thay đổi
  - `$data = $category->all(array('name', 'id'));` Truy vấn dữ liệu trong Categories
  - `foreach ($data as $list){...}` Duyệt qua tập kết quả và hiển thị lên trình duyệt
- ✓ Chạy trên trình duyệt và xem kết quả (phụ thuộc dữ liệu thực do sinh viên nhập)

```

1 5 CLOTHING
2 4 FASHION
3 16 HEAVY METAL
4 3 KIDS
5 1 MENS
6 2 WOMENS

```

- ✓ Eloquent cho phép xóa record thông qua lệnh `$category->delete();`. Ví dụ xóa CLOTHING, đầu tiên mở `/app/Http/routes.php` và thêm code:

```
1 Route::get('/delete', function() {  
2     $category = App\Category::find(5);  
3     $category->delete();  
4  
5     $data = $category->all(array('name', 'id'));  
6  
7     foreach ($data as $list) {  
8         echo $list->id . ' ' . $list->name . '  
9     }  
10 }));
```

✓ Chạy và kiểm tra kết quả

```
1 | http://localhost/larashop/public/delete
```

#### **BÀI 4 (2 ĐIỂM) : Giảng viên tự cho thêm**