**GIÁO TRÌNH LARAVEL - BUỔI 14**

**RELATIONSHIP**

1. **Cấu truc tổng quát**

|  |
| --- |
| public function tenlk() {  return $this->phuong\_thuc('model\_muon\_lk',…..v.v);  } |

1. Định nghĩa

Eloquent relationships được định nghĩa như một hàm trong class Eloquent model. Vì vậy, giống như là Eloquent mô tả chính mình, các quan hệ cũng phục vụ rất mạnh mẽ bởi [query builders](http://laravel.com.vn/docs/5.7/queries), định nghĩa các relationshipsnhư các hàm cũng cấp các ràng buộc mạnh mẽ và khả năng truy vấn

-Trước đây Laravel version 4.0 thì chỉ cung cấp cho chúng ta 3 mối quan hệ chính, nhưng đến phiên bản hiện tại con số đó đã được nâng lên là **6**mối quan hệ.

-Gồm:

* **One to One** : Một Một.
* **One to Many**: Một nhiều.
* **Many to Many**: Nhiều nhiều.
* **Has Many Through**: Nhiều nhiều qua lại trung gian.
* **Polymorphic Relations**: Đa hình.
* **Many To Many Polymorphic Relations** : Nhiều nhiều đa hình.

1. **Quan hệ one one**

Với quan hệ one one ta có những phương thức liên kết sau:

* + HasOne (dùng để liên kết từ bảng chính tới bảng phụ)
  + BelongsTo (dùng để liên kết từ bảng phụ tới bảng chính)

Bảng chính: là bảng chứa khoá chính.

Bảng phụ: là bảng chứa khoá ngoại.

Ví Dụ:

Bảng users

Chi tiết bảng: id là khoá chính.

|  |  |  |  |
| --- | --- | --- | --- |
| Users | | | |
| id | email | password | level |
| 1 | [A@gmail.com](mailto:A@gmail.com) | 123456 | 1 |
| 2 | [B@gmail.com](mailto:B@gmail.com) | 123456 | 2 |
| 3 | [C@gmail.com](mailto:C@gmail.com) | 123456 | 1 |
| 4 | [D@gmail.com](mailto:D@gmail.com) | 123456 | 2 |

Bảng info

Chi tiết bảng: id là khoá chính và users\_id là khoá ngoại liên kết với bảng users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Info | | | | |
| id | id\_number | address | phone | users\_id |
| 1 | 174444 | Thường tín | 4444444 | 4 |
| 2 | 174222 | Hưng yên | 2222222 | 2 |
| 3 | 174111 | Nghệ an | 1111111 | 1 |
| 4 | 174333 | Thanh hoá | 3333333 | 3 |

Ví dụ: Chúng ta có bảng users và bảng info (1 thành viên chỉ có 1 thông tin cá nhân và ngược lại) . đây là quan hệ one one

Cách khai báo:

|  |
| --- |
| public function tên\_liên\_kết()  {  return $this->hasOne('gọi model liên kết bằng namespace', 'Khoá ngoại', 'khoá chính');  } |

Do chúng ta có 2 phương thức liên kết lên có 2 trường hợp có thể xảy ra:

* Trường hợp 1

Từ 1 bản ghi trong bảng users ta sẽ lấy ra thông tin( bảng info) tương ứng với thành viên đó

Đây là liên kết từ bảng chính (bảng users chứa khoá chính) tới bảng phụ (bảng info chứa khoá ngoại liên kết đến bảng users)

Trong trường hợp này chúng ta dùng hasOne

Trong model users

|  |
| --- |
| class users extends Model  {  protected $table='users'; // liên kết đến bảng users  public $timestamps =false; //bảng users không sử dụng timestamps  // tạo liên kết để lấy dữ liệu từ bảng info    public function info()  {  return $this->hasOne('App\info', 'users\_id', 'id');  }  } |

Câu lệnh lấy dữ liệu qua liên kết

*\*\*Chú ý:*

* *Muốn sử dụng model thì chúng ta phải gọi model thông qua namespace:*

*Use App\users;*

|  |
| --- |
| $data = users::find(2)->info->get()->toArray();  // toArray chuyển từ dạng object sang array |

* Trường hợp 2

Từ 1 bản ghi trong bảng info ta sẽ lấy ra thành viên có thông tin đó. Đây là liên kết từ bảng phụ tới bảng chính

Cấu trúc khai báo liên kết tương tự như từ bảng chính tới bảng phụ

Trong trường hợp này chúng ta dùng

|  |
| --- |
| public function user()  {  return $this->belongsTo('App\Info, 'users\_id', 'id');  } |

Câu lệnh lấy dữ liệu qua liên kết

|  |
| --- |
| $data = info::find(2)->user->get()->toArray(); |

1. **One Many**

Cấu trúc tổng quát:

|  |
| --- |
| public function Tên\_liên\_kết()  {  return $this->hasMany(' gọi model liên kết bằng namespace ', 'khoá\_ngoại, 'khoá chính');  } |

Ví dụ: chúng ta có bảng Thành viên(users) và comment(1 thành viên thì có nhiều comment và 1 comment chỉ thuộc 1 thành viên mà thôi) đây là quan hệ 1 nhiều

Bảng user

Chi tiết bảng: id là khoá chính và users\_id là khoá ngoại liên kết với bảng users.

|  |  |  |  |
| --- | --- | --- | --- |
| Users | | | |
| id | email | password | level |
| 1 | [A@gmail.com](mailto:A@gmail.com) | 123456 | 1 |
| 2 | [B@gmail.com](mailto:B@gmail.com) | 123456 | 2 |
| 3 | [C@gmail.com](mailto:C@gmail.com) | 123456 | 1 |
| 4 | [D@gmail.com](mailto:D@gmail.com) | 123456 | 2 |

Bảng comment

Chi tiết bảng: id là khoá chính và users\_id là khoá ngoại liên kết với bảng users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Comment | | | | |
| id | content | users\_id | created\_at | updated\_at |
| 1 | Xin chào! | 2 |  |  |
| 2 | Tôi tên là B! | 2 |  |  |
| 3 | Các bạn Tên gì vậy? | 2 |  |  |
| 4 | Tôi tên A. | 1 |  |  |
| 5 | Còn ai không | 1 |  |  |
| 6 | Còn Tôi tên C | 3 |  |  |

Trong model users

public function comment()

{

return $this->hasMany('App\Comment', 'users\_id', 'id');

}

Để sử dụng liên kết

|  |
| --- |
| $data = users::find(2)->comment()->get()->toArray();  //$data sẽ được 1 mảng chứ nhiều bản ghi |

*\*\*Chú ý:*

* *Nếu muốn từ 1 bản ghi trong comment chúng ta lấy ra thành viên comment thì bài toán thành liên kết 1-1 (trường hợp 2) . Do 1 comment chỉ tương ứng với 1 thành viên mà thôi*

1. **Many-to-Many**

Nhắc đến liên kết 1-n thì người ta nghĩ ngay đến việc tạo bảng trung gian chứa 2 khoá ngoại liên kết đến 2 bảng chứa khoá chính .

Cách khai báo:

|  |
| --- |
| public function tên\_liên\_kết()  {  return $this->belongsToMany('' gọi model muốn liên kết bằng namespace '', 'tên bảng trung gian ‘, 'khoá ngoại liên kết đến bảng cần lk ', 'khoá ngoại liên kết đến bảng trong class');  } |

Ví dụ: chúng ta có bảng thành viên và bảng môn học(1 thành viên có thể đăng ký nhiều môn học và 1 môn có nhiều thành viên đăng ký) đây là quan hệ nhiều nhiều

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Users | | | | | |
| id | email | | | password | level |
| 1 | [A@gmail.com](mailto:A@gmail.com) | | | 123456 | 1 |
| 2 | [B@gmail.com](mailto:B@gmail.com) | | | 123456 | 2 |
| 3 | [C@gmail.com](mailto:C@gmail.com) | | | 123456 | 1 |
| 4 | [D@gmail.com](mailto:D@gmail.com) | | | 123456 | 2 |
| Users\_Subject | | |
| users\_id | | subjects\_id |
| 1 | | 1 |
| 1 | | 2 |
| 1 | | 3 |
| 2 | | 2 |
| 2 | | 4 |

Bảng class

|  |  |
| --- | --- |
| subjects | |
| **id** | **name** |
| **1** | Toán |
| **2** | Lý |
| **3** | Hoá |
| **4** | Văn |

Chúng ta có thêm bảng trung gian users\_subjects chứa khoá ngoại của cả 2 bảng

Trong model users

|  |
| --- |
| public function subjects()  {  return $this->belongsToMany('App \ subjects’, ‘users\_ subjects ', ' subjects \_id', 'users\_id');  } |

Sử dụng liên kêt:

|  |
| --- |
| $data = users::find(2)-> subjects ()->get()->toArray();  //$data nhận được nhiều bàn ghi |

Trường hợp từ 1 môn học tìm ra các thành viên đăng ký môn đó thì chúng ta chỉ việc thay đổi 1 chút.

Trong model subjects

|  |
| --- |
| public function users()  {  return $this->belongsToMany('App \users’, ‘users\_ subjects ', 'users\_id', ' subjects \_id');  } |

Sử dụng liên kêt:

|  |
| --- |
| $data = subjects::find(2)-> users ()->get()->toArray();  //$data nhận được nhiều bàn ghi |