**Bài thực hành 07**

**Eloquent ORM**

***Mục tiêu***

* Sử dung các truy vấn nâng cao như: Join 1 bảng , nhiều bảng
* Sử dụng các thành phần FormRequest để validate form
* Tạo model và thực hiện các truy ván dữ liệu thông qua model

**Thực hành theo các bài tập**

Xét quan hệ giữa 2 thực thể category và product

***Chú ý: Có thể kế thừa CSDL từ các lab trước***



Laravel cung cấp các phương thứ JOIN như

hasOne , hasMany JOIN gữa 2 bảng 1-n

belongsToMany để JOIN 2 bảng n-n thông qua bảng trung gian

Cú pháp chung

**hasOne**

*/\*\**

*\* hasOne là phương thức join 1-1 ví dự từ 1 product - 1 ategory*

*\* ModelName là Model đang cần join tới VD Category*

*\* @primary\_key Là khóa chính của ModelName VD id của category*

*\* @foreign\_key là khóa ngoại tham chiếu VD category\_id của product*

*\**

*\*/*

public function func\_name(){

    return $this->hasOne(ModelName::class, 'primary\_key', 'foreign\_key');

}

**hasMany**

*/\*\**

*\* hasMany là phương thức join 1-n ví dự từ 1 ategory - n product*

*\* ModelName là Model đang cần join tới VD Product*

*\* @foreign\_key là khóa ngoại tham chiếu VD category\_id của product*

*\* @primary\_key Là khóa chính của ModelName VD id của category*

*\**

*\*/*

public function func\_name(){

    return $this->hasMany(ModelName::class, 'foreign\_key', 'primary\_key');

}

**belongsToMany**

*/\*\**

*\* belongsToMany là phương thức join n-n ví dự từ n product - n color*

*\* thông qua bảng trung gian là product\_color*

*\* ModelName là Model đang cần join tới*

*\* @table\_name là bảng trung gian VD product\_color*

*\* @foreign\_key1 là khóa ngoại tham chiếu tới bảng cảu ModelName VD product\_id tham chiếu tới product*

*\* @foreign\_key2 Là khóa ngoại của Model còn lại VD color\_id tới color*

*\*/*

public function func\_name(){

    return $this->belongsToMany(ModelName::class,'table\_name' 'foreign\_key1', 'foreign\_key2');

}

# Bài 1: Tìm hiểu câu lệnh JOIN 2 bảng có quan hệ

**Bước 1: Tạo cấu trúc hai bảng như sau**

CREATE TABLE categories (

    id INT PRIMARY KEY AUTO\_INCREMENT,

    name VARCHAR(100) NOT NULL UNIQUE,

    status tinyint DEFAULT '1',

    created\_at date DEFAULT NOW(),

    updated\_at date NULL

);

CREATE TABLE IF NOT EXISTS products (

  id INT PRIMARY KEY AUTO\_INCREMENT,

  name VARCHAR(150) NOT NULL,

  price float NOT NULL,

  sale\_price float NOT NULL,

  image VARCHAR(150) NOT NULL,

  content text null,

  category\_id INT NOT NULL,

  status INT DEFAULT '0',

  created\_at date DEFAULT NOW(),

  updated\_at date NULL,

  FOREIGN KEY (category\_id) REFERENCES categories (id)

);

Sau đó thêm mới dữ liệu 2 bảng cho phù hợp

**Bước 2:** Tạo model Category và khai báo phương thức JOIN tới bảng product để lấy danh sách sản phẩm của danh mục

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Category extends Model

{

    protected $table = 'category';

    protected $fillable = ['name','status','parent\_id'];

    public $timestamps = false; *//loại bỏ 2 cột created\_at và updated\_at*

*// phương thức join tới bảng product lấy ra mảng Product*

    public function products(){

        return $this->hasMany(Product::class, 'category\_id', 'id');

    }

}

**Bước 3:** Tạo model Product join tới Category để lấy danh mục của sản phẩm

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model

{

    protected $table = 'product';

    protected $fillable = ['name','status','category\_id','price','image','descriptions'];

    public $timestamps = false; *//loại bỏ 2 cột created\_at và updated\_at*

*// phương thức join tới bảng category*

    public function cat(){

        return $this->hasOne(Category::class, 'id', 'category\_id');

    }

}

**Bước 4: Tạo Route, controller và view danh sách Product**

**Route routes/web.php)**

*/\*\* Trang chủ \*/*

Route::get('/', 'HomeController@index')->name('home');

*/\*\* Danh sách sản phẩm link http://localhost:port/project\_folder/product \*/*

Route::get('/product','ProductController@index')->name('product');

**Controller app\Http\Controllers\ProductController**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;

class ProductController extends Controller

{

    public function index(){

        $products = Product::paginate(15);

        return view('product',compact('products'));

    }

}

**View resources/views/product.blade.php**

@extends('layouts.main')

@section('main')

<div class="product-box">

    <table class="table table-hover">

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Category Name</th>

                <th>Price</th>

            </tr>

        </thead>

        <tbody>

            @foreach($products as $product)

            <tr>

                <td>{{$product->id}}</td>

                <td>{{$product->name}}</td>

*<!-- Hiển thị tên danh mục theo phương thức JOIN đã hasOne trong Product model -->*

                <td>{{$product->cat->name}}</td>

                <td>{{$product->price}}</td>

            </tr>

            @endforeach

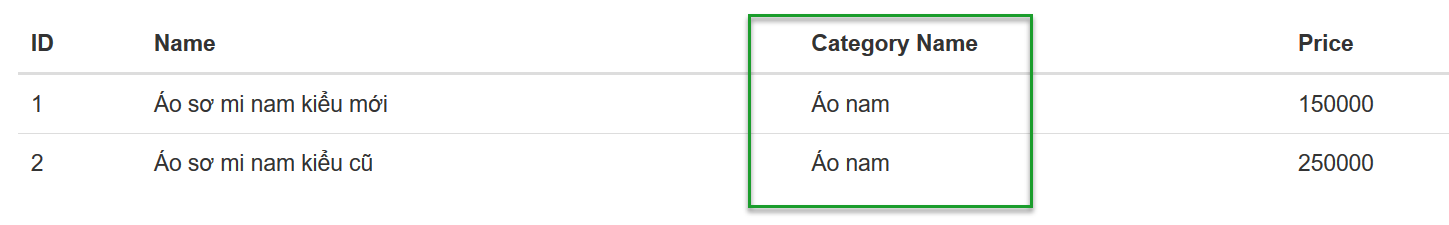
        </tbody>

    </table>

</div>

@stop()

**Hãy truy cập thử:** [**http://localhost:port/project\_folder/product**](http://localhost:port/project_folder/product)



**Bước 5: Tạo route, controller, view chi tiết danh mục Category**

**Route (routes/web.php)**

*/\*\* Chi tiết danh mục theo id link http://localhost:port/project\_folder/category/1\*/*

Route::get('/category/{id}','CategoryController@detail')->name('category');

**Controller app\Http\Controllers\CategoryController**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Category;

class CategoryController extends Controller

{

    public function detail($id){

        $cat = Category::find($id);

        return view('category', compact('cat'));

    }

}

**View resources/views/category**

@extends('layouts.main')

@section('main')

<div class="category-box">

    <h2>Name: {{$cat->name}}</h2>

*<!-- duyệt danh sách sản phẩm của cateo=gory theo phương thức đã*

*hasMany trong Caegory model -->*

    <h3>Product List</h3>

    <table class="table table-hover">

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

                <th>Price</th>

            </tr>

        </thead>

        <tbody>

*<!-- duyệt danh sách sản phẩm của cateo=gory theo phương thức đã*

*hasMany trong Caegory model -->*

            @foreach($cat->products as $product)

            <tr>

                <td>{{$product->id}}</td>

                <td>{{$product->name}}</td>

                <td>{{$product->price}}</td>

            </tr>

            @endforeach

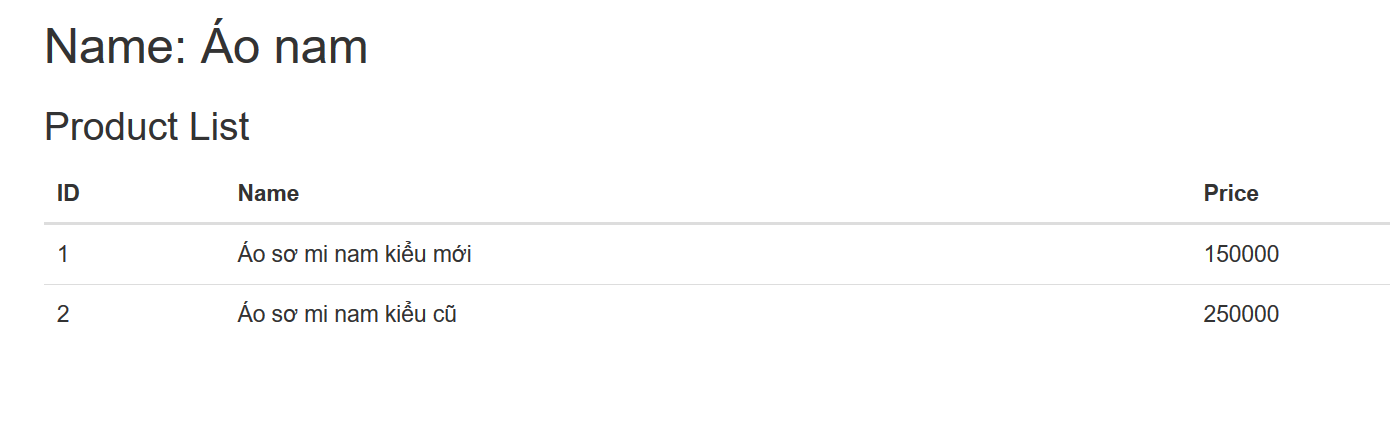
        </tbody>

    </table>

</div>

@stop()

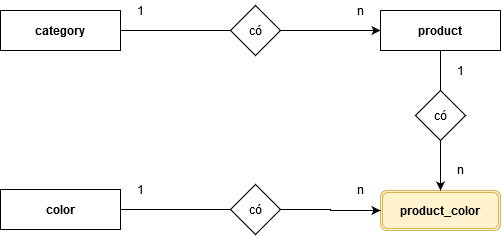
**Truy cập link** [**http://localhost:port/project\_folder/category/1**](http://localhost:port/project_folder/category/1)



**Bài 2: Sử dụng belongsToMany join 2 bảng**

**Khi có quan hệ n-n VD giữa bảng product và bảng color, 1 product có thể có n color và 1 color có thể sử dụng cho nhiều product**

**Khi đó sinh ra thực thể yếu là product\_color như sơ đồ quan hệ dưới đây**



**Vấn đề đặt ra: muốn lấy danh sách color của một sản phẩm, khi đó sử dụng belongsToMany**

**Bước 1: Tạo thêm bảng color và thực thể yếu product\_color với các khóa ngoại như sau**

CREATE TABLE IF NOT EXISTS color (

  id INT PRIMARY KEY AUTO\_INCREMENT,

  name VARCHAR(100) NOT NULL UNIQUE

);

CREATE TABLE IF NOT EXISTS product\_color (

  product\_id INT NOT NULL,

  color\_id INT NOT NULL,

  PRIMARY KEY (product\_id, color\_id), *-- khóa chính trên 2 cột*

  FOREIGN KEY (product\_id) REFERENCES product (id),

  FOREIGN KEY (color\_id) REFERENCES color (id)

);

**Hãy thêm dữ liệu vào các bảng cho phù hợp**

**Bước 2: Tạo thêm model Color và ProductColor**

**Model Color**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Color extends Model

{

    protected $table = 'color';

    protected $fillable = ['name'];

    public $timestamps = false; *//loại bỏ 2 cột created\_at và updated\_at*

}

**Model ProductColor**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class ProductColor extends Model

{

    protected $table = 'product\_color';

    protected $fillable = ['product\_id','color\_id'];

    public $timestamps = false; *//loại bỏ 2 cột created\_at và updated\_at*

}

**Bước 3: Mở lại Product model và khai báo phương thức lấy ra danh sách color của product**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model

{

    protected $table = 'product';

    protected $fillable = ['name','status','category\_id','price','image','descriptions'];

    public $timestamps = false; *//loại bỏ 2 cột created\_at và updated\_at*

*// phương thức join tới bảng category*

    public function cat(){

        return $this->hasOne(Category::class, 'id', 'category\_id');

    }

*// phương thức join tới bảng color thông qua bảng product\_color*

    public function colors(){

        return $this-> belongsToMany (Color::class, 'product\_color','product\_id','color\_id');

    }

}

**Bước 4: Tạo route, controller action, view chi tiết sản phẩm**

**Route**

Route::get('/product/{id}','ProductController@detail')->name('product\_detail');

**Controller (Mở lại ProductController thêm phương thưc detail )**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;

class ProductController extends Controller

{

    public function index(){

        $products = Product::paginate(15);

        return view('product',compact('products'));

    }

     public function detail($id){

        $product = Product::find($id);

        return view('product\_detail', compact('product'));

    }

}

**View resources/views/product\_detail.blade.php**

@extends('layouts.main')

@section('main')

<div class="category-box">

    <h2>Name: {{$product->name}}</h2>

    <h2>Price: {{$product->price}}</h2>

    <h2>Category nane: {{$product->cat->name}}</h2>

*<!-- duyệt danh sách màu của product theo phương thức đã belongsToMany trong Product model -->*

    <h3>Color List</h3>

    <table class="table table-hover">

        <thead>

            <tr>

                <th>ID</th>

                <th>Name</th>

            </tr>

        </thead>

        <tbody>

*<!-- duyệt danh sách colors của product theo phương thức đã belongsToMany trong Product model -->*

            @foreach($product->colors as $color)

            <tr>

                <td>{{$color->id}}</td>

                <td>{{$color->name}}</td>

            </tr>

            @endforeach

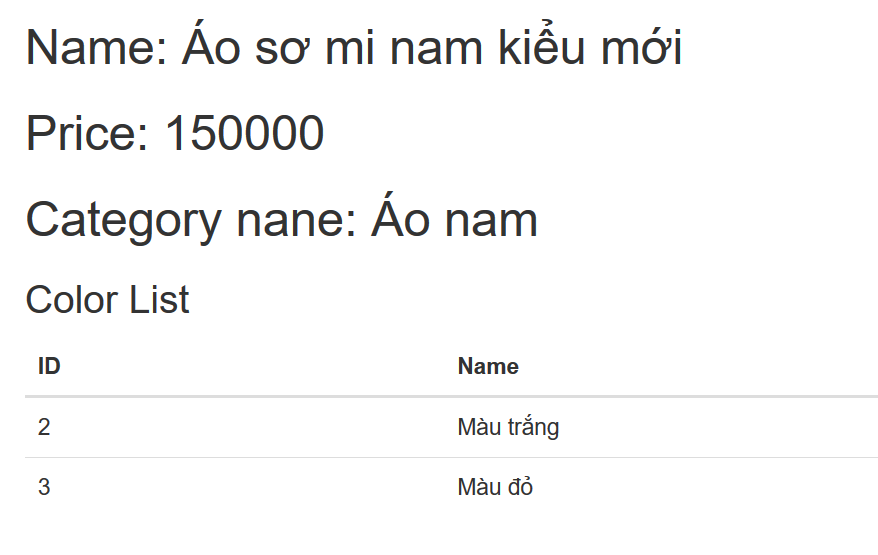
        </tbody>

    </table>

</div>

@stop()

**Truy cập link** [**http://localhost:port/project\_folder/product/1**](http://localhost:port/project_folder/product/1)



**Bài 3: Tạo các local scope để truy vấn**

Có thể tạo thêm các phương thức truy vấn cho model để tiện truy vấn

Bước 1: tạo scopeSearch truy vấn tìm kiếm

Mở model Product và khai báo scopeSearch

public function scopeSearch($query){

    if(request()->key){

        $key = request()->key;

        $query = $query->where('name','LIKE','%'.$key.'%);

    }

    return $query;

}

**Bước 2: Tại ProductController@index khai báo lại truy vấn**

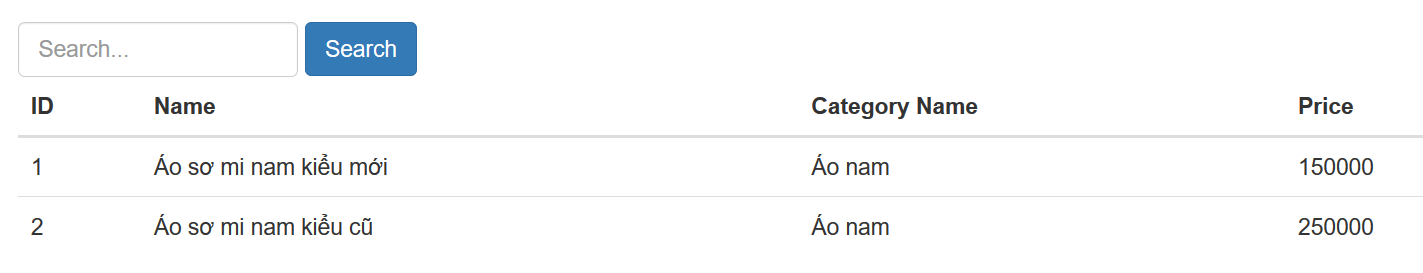
public function index(){

    $products = Product::search()->paginate(15);

    return view('product',compact('products'));

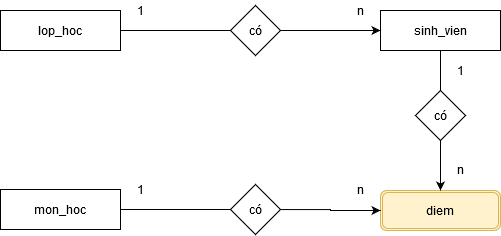
}

**Bước 3: Tại view product.blade.php tạo form tìm kiến**



**Bài tập tự làm**

**Yêu cầu: Thiết kế CSDL quan\_ly\_sinh\_vien gồm các thực thể sau**

**Ye**

**Yêu cầu 2: Tạo Các model: LopHoc, SinhVien, Diem, MonHoc**

Yêu cầu 3: Thực hiện các chức năng danh sách, thêm sửa xóa trên các bảng trên, dữ liệu được sắp xếp giảm dần theo id

**Yêu cầu 4: Khai báo các phương thức hasOne, hasMany, belongsToMany trong các model cho phù hợp để thực hiện các yêu cầu sau**

1. Khi xem chi tiết lớp học hiển thị danh sách sinh viên của lớp đó
2. Khi xem chi tiết sinh viên xem được điểm các môn học cảu sinh viên đó

**Yêu cầu 5: Tạo thêm scopeSearch theo tên sinh viên và form tìm kiếm**