

BUỔI 11:

ELOQUENT MODEL TRONG LARAVEL

Laravel 8 cung cấp 2 kiểu truy vấn phổ biến với database là **Eloquent ORM** và **Query Builder**. Nếu như Query Builder làm việc trực tiếp với database thông qua các truy vấn thì Eloquent ORM lại sử dụng Model để tương tác với Database.

11.1/ Eloquent ORM là gì?

- ORM(Object Relational Mapping) đây là tên gọi chỉ việc ánh xạ các record dữ liệu trong hệ quản trị cơ sở dữ liệu sang dạng đối tượng mà mã nguồn đang định dạng trong class.
- Eloquent ORM: Là một giải pháp tương tác với cơ sở dữ liệu sử dụng kỹ thuật ORM giúp lập trình viên thao tác dễ dàng hơn với DB.
- Eloquent Model (Model) - là model thao tác trực tiếp với DB, xử lý logic nghiệp vụ và trả về dữ liệu cho controller.
- Eloquent ORM trong Laravel 8 sử dụng các Model để tương tác với cơ sở dữ liệu, thông thường mỗi bảng của database sẽ được ánh xạ qua 'Model', và model này được sử dụng để tương tác với bảng.

11.2/ Tạo Model

Cú pháp: **php artisan make:model <model_name>**

Trong đó: **ModelName** là tên của model ta muốn tạo.

Ví dụ: tạo mới một model tên là Monhoc

```
PS D:\H@oCon\Data\HocLaravel\thlaravel> php artisan make:model Monhoc  
Model created successfully.  
PS D:\H@oCon\Data\HocLaravel\thlaravel> █
```

Kết quả file **Monhoc.php** sẽ được tạo ra trong thư mục **app/Models** có code như sau:

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Monhoc extends Model  
{  
    use HasFactory;  
}
```

11.3/ Cấu hình Model

11.3.1/ Định nghĩa bảng vào Model

Cú pháp: **protected \$table = 'table_name';**

Ví dụ: định nghĩa bảng **monhoc** cho model **Monhoc**

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Monhoc extends Model
{
    use HasFactory;
    protected $table = 'monhoc';
}

```

11.3.2/ Khai báo khóa chính

Cú pháp: **protected \$primaryKey = 'field_name';**

Ví dụ: khai báo khóa chính của bảng **monhoc** cho model **Monhoc**

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Monhoc extends Model
{
    use HasFactory;
    protected $table = 'monhoc';
    protected $primaryKey = 'mamh';
}

```

Mặc định, Laravel sẽ coi như trong table của bạn có 2 column **created_at** và **deleted_at**, và sẽ mặc định set giá trị cho 2 cột dữ liệu này cho các trường hợp tạo mới, update dữ liệu trong bảng. Nếu bạn không muốn Laravel làm việc này bạn có thể thiết lập thuộc tính **\$timestamps** về **false**.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Monhoc extends Model
{
    use HasFactory;
    protected $table = 'monhoc';
    protected $primaryKey = 'mamh';
    public $timestamps = false;
}

```

11.4/ Các câu truy vấn hay dùng trong Eloquent ORM.

11.4.1/ Lấy tất cả dữ liệu trong bảng

Cú pháp: `ModelName::all();`

Ví dụ: Hiển thị danh sách các môn học

Bước 01: Tạo View tên là **list_subjects.blade.php** với code như sau:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Danh Sách Môn Học</title>
</head>
<body>
    <h1 align="center">DANH SÁCH MÔN HỌC</h1>
    <table class="table table-light">
        <thead class="thead-light">
            <tr>
                <th>Mã Môn Học</th>
                <th>Tên Môn Học</th>
                <th>Số Tín Chỉ</th>
            </tr>
        </thead>
        <tbody>
            @foreach($subjects as $subject)
                <tr>
                    <td>{{ $subject->mamh }}</td>
                    <td>{{ $subject->tenmh }}</td>
                    <td>{{ $subject->sotinch }}</td>
                </tr>
            @endforeach
        </tbody>
        <tfoot>
            <tr>
                <th colspan="3">Tổng số môn học là: {{ $total }}</th>
            </tr>
        </tfoot>
    </table>
</body>
</html>
```

Bước 02: Mở tập tin mycontroller.php thêm phương thức như sau:

```
public function display_all_subjects()
{
    $subjects = Monhoc::all();
    $total = Monhoc::all()->count();
    return view('Buoi11.list_subjects', compact("subjects", "total"));
}
```

Bước 03: Mở tập tin web.php thêm route như sau:

```
//Buổi 11
Route::get("/lietkedsmh", [mycontroller::class, "display_all_subjects"]);
```

11.4.2/ Lấy 1 mẫu tin thông qua khóa chính trong bảng

Cú pháp: `ModelName::find(<value>);`

Ví dụ: Tìm môn học có giá trị khóa chính là 10

```
$subject = Monhoc::find(10);
```

11.4.3/ truy vấn điều kiện

Cú pháp: `ModelName::where('fieldname', 'operator', 'value')->get();`

Trong đó: operator có thể là = > >= < <= <>

11.4.4/ truy vấn chọn cột dữ liệu

Cú pháp: `ModelName::select('fieldname1', 'fieldname2', ...)->get();`

11.4.5/ đếm dữ liệu

Cú pháp: `ModelName::all()->count();`

11.4.6/ thêm dữ liệu

```
$variable_name = new ModelName();
$variable_name->fieldname1 = "value";
...
$variable_name->fieldnamen = "value";
$variable_name->save();
```

11.4.7/ Sửa dữ liệu

Cách 01:

```
$var_name = ModelName::find(<value>);
$var_name->fieldname1 = "value";
...
$var_name->save();
```

Cách 02:

```
ModelName::where('pk', <value>)->update(['fieldname1' => 'value',...]);
```

11.4.8/ Xóa dữ liệu

Cách 01:

```
$var_name= ModelName::find(value);
$var_name->delete();
```

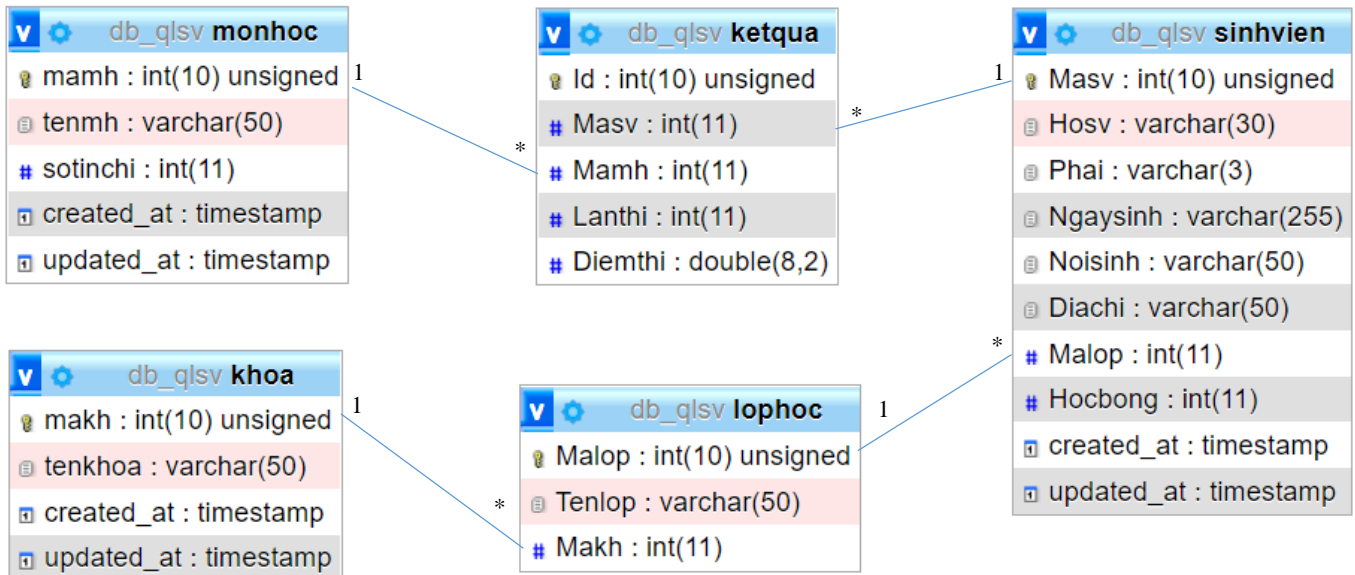
Cách 02:

```
ModelName::destroy(value);
```

Cách 03:

```
ModelName::where('fieldname', value)->delete();
```

11.5/ Relationship trong Eloquent Model.



11.5.1/ Bảng Khoa và Lophoc.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Khoa extends Model
{
    use HasFactory;
    protected $table = 'khoa';
    protected $primaryKey = 'makh';
    public $timestamps = false;

    public function lophoc()
    {
        return $this->hasMany(Lophoc::class, "Makh", "makh");
    }
}
```

Một Khoa có nhiều lớp học

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Lophoc extends Model
{
    use HasFactory;
    protected $table = 'lophoc';
    protected $primaryKey = 'Malop';
    public $timestamps = false;

    public function Khoa()
    {
        return $this->belongsTo(Khoa::class, "Makh", "makh");
    }

    public function Sinhvien()
    {
        return $this->hasMany(Sinhvien::class, "Malop", "Malop");
    }
}
```

Mỗi lớp học chỉ thuộc một khoa

Một lớp có nhiều sinh viên

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Sinhvien extends Model
{
    use HasFactory;
    protected $table = 'sinhvien';
    protected $primaryKey = 'Masv';
    public $timestamps = false;

    public function Lophoc()
    {
        return $this->belongsTo(Lophoc::class, "Malop", "Malop");
    }

    public function Ketqua()
    {
        return $this->hasMany(Ketqua::class, "Masv", "Masv");
    }
}
```

Mỗi sinh viên chỉ thuộc một lớp học

Một sinh viên có nhiều kết quả thi

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Monhoc extends Model
{
    use HasFactory;
    protected $table = 'monhoc';
    protected $primaryKey = 'mamh';
    public $timestamps = false;

    public function Ketqua()
    {
        return $this->hasMany(Ketqua::class, "Mamh", "Mamh");
    }
}
```

Một môn học có nhiều sinh viên và có nhiều lần thi cho mỗi sinh viên

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Ketqua extends Model
{
    use HasFactory;
    protected $table = 'ketqua';
    protected $primaryKey = 'Id';
    public $timestamps = false;

    public function Sinhvien()
    {
        return $this->belongsTo(Sinhvien::class, "Masv", "Id");
    }


    public function Monhoc()
    {
        return $this->belongsTo(Monhoc::class, "Mamh", "Id");
    }
}
```

Kết quả của mỗi lần thi chỉ thuộc một sinh viên





















Kết quả của mỗi lần thi chỉ có một môn thi

Bài Tập Ứng Dụng

Xây dựng trang Web có mẫu như sau:


TTĐTBK

Môn Học ▾ Khoa ▾ Sinh viên ▾ Kết quả ▾

Mã Môn Học	Tên Môn Học	Số Tín Chỉ	Thao tác
1	Trí tuệ nhân tạo	4	 
2	Truyền tin	4	 
3	Đồ họa	8	 
4	Văn Phạm	7	 
5	Đàm thoại	5	 
6	Đồ họa	8	 
7	Vật lý	4	 
8	Vật lý đại cương	4	 
9	Triết học	6	 
10	Toán đại cương	4	 
Tổng số môn học: 10			

Yêu cầu:

- Khi người dùng chọn **Môn học >> Thêm Mới** thì xuất hiện giao diện:



TTĐTBK

Môn Học ▾ Khoa ▾ Sinh viên ▾ Kết quả ▾

Thêm môn học mới

Tên môn học

Số Tín Chỉ

- Khi người dùng bấm vào  thì xuất hiện giao diện sửa thông tin môn học.
- Khi người dùng bấm vào  thì sẽ xóa môn học.

Hướng dẫn:

Bước 01: Tạo thư mục Buoi07 vào Views và tạo trang chung tên là master.blade.php vào Buoi07 với code như sau:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <!-- Bootstrap CSS -->
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
<title>Trang Chủ</title>
<base href="{{asset('/') }}">
</head>
<body>
<div class="container mb-4">
  <div class="float-start">
     TTĐTBK
  </div>
  <form class="d-flex float-end p-2">
    <input class="form-control me-2" type="search" placeholder="Nhập từ khóa" aria-
label="Search">
    <button class="btn btn-outline-success" type="submit"> Tìm </button>
  </form>
  <ul class="nav nav-pills justify-content-center p-2">
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#">Môn Học</a>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item" href="{{route('lkmh')}}">Liệt kê</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item" href="{{route('thmh')}}">Thêm mới</a></li>
      </ul>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#">Khoa</a>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item">Liệt kê</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item">Thêm mới</a></li>
      </ul>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#">Sinh viên</a>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item">Liệt kê</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item">Thêm mới</a></li>
      </ul>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#">Kết quả</a>
      <ul class="dropdown-menu">
        <li><a class="dropdown-item">Liệt kê</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item">Thêm mới</a></li>
      </ul>
    </li>
  </ul>
</div>
<div class="container mt-1 bg-info" style="clear:both">
  @yield("noidung")

```



```

</div>
<!-- Option 1: Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>

```

Bước 02: Mở tập tập Web.php thêm route như sau:

```

//Buổi 07
Route::group(["prefix" => "Buoi07"], function(){
    Route::group(["prefix" => "monhoc"],function(){
        Route::get("lietkemonhoc", [baitapcontroller::class, "get_lkmonhoc"])->name("lkmh");
        Route::get("themmonhoc", [baitapcontroller::class, "get_themmonhoc"])->name("thmh");
        Route::post("themmonhoc", [baitapcontroller::class, "post_themmonhoc"])->name("ltmh");
        Route::get("suamonhoc/{mamh}", [baitapcontroller::class, "get_suamonhoc"])->name('suamh');
        Route::post("suamonhoc/{mamh}", [baitapcontroller::class, "post_suamonhoc"])->name('cnmh');
        Route::get("xoamonhoc/{mamh}", [baitapcontroller::class, "get_xoamonhoc"])->name("xoamh");
    });
});

```

Bước 03: Tại View tên là themmonhoc.blade.php với code như sau:

```

@extends("buoi07.master")
@section("noidung")
    <div class="h2 alert alert-info text-center">Thêm môn học mới</div>
    @if($errors->any())
        <div class="alert alert-danger">
            <strong>Whoops</strong> Bậnhập dữ liệu không hợp!!!<br><br>
        </div>
        <ul>
            @foreach($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    @endif
    <form action="{{route('ltmh')}}" method="post">
        @csrf
        <div class="form-group">
            <strong>Tên môn học</strong>
            <input type="text" name="tenmh" class="form-control" value="">
        </div>
        <div class="form-group">
            <strong>Số Tín Chỉ</strong>
            <input type="number" name="sotinchi" class="form-control" value="">
        </div>
        <div class="form-group my-2">
            <input type="submit" name="save" class="btn btn-primary" value=" Lưu Trữ "> |
            <a href="{{route('lkmh')}}" class="btn btn-warning">Trở về</a>
        </div>
    </form>
@endsection

```

Bước 04: Tạo view suamonhoc.blade.php với code như sau:

```
@extends("Buoi07.master")
@section("noidung")
    <div class="h2 alert alert-info text-center">Cập nhật môn học</div>
    <form action="{{route('cnmh',$monhoc->mamh)}}" method="post">
        @csrf
        <div class="form-group">
            <strong>Tên môn học</strong>
            <input type="text" name="tenmh" class="form-control" value="{{ $monhoc->tenmh }}">
            @if ($errors->has("tenmh"))
                <span class="text-danger">{{ $errors->first("tenmh") }}</span>
            @endif
        </div>
        <div class="form-group">
            <strong>Số Tín Chỉ</strong>
            <input type="number" name="sotinchi" class="form-control" value="{{ $monhoc->sotinchi }}">
            @if ($errors->has("sotinchi"))
                <span class="text-danger">{{ $errors->first("sotinchi") }}</span>
            @endif
        </div>
        <div class="form-group my-2">
            <input type="submit" name="save" class="btn btn-primary" value=" Cập Nhật ">|
            <a href="{{route('lkmh')}}" class="btn btn-warning"> Trở về </a>
        </div>
    </form>
@endsection
```

Bước 05: Mở tập baitapcontroller.php và thêm các phương thức như sau:

```
//Buổi 7
public function get_lkmonhoc()
{
    $dsmh = Monhoc::all();
    return view("buoi07.monhoc", compact('dsmh'));
}

public function get_themmonhoc()
{
    return view("buoi07.themmonhoc");
}

public function post_themmonhoc(Request $req)
{
    $vl = $req->validate([
        'tenmh' => 'required|min:7',
        'sotinchi' => 'required|numeric|min:3|max:10'
    ], [
        'tenmh.required' => 'Bạn chưa nhập tên môn học',
        'tenmh.min' => 'Tên môn học tối thiểu 7 ký tự',
        'sotinchi.required' => 'Chưa nhập số tín chỉ',
        'sotinchi.numeric' => 'phải nhập số',
        'sotinchi.min' => 'tối thiểu 3 tín chỉ',
        'sotinchi.max' => 'tối đa 10 tín chỉ'
    ]);
}
```

```

    $monhoc = new Monhoc();
    $monhoc->tenmh = $v1["tenmh"];
    $monhoc->sotinchih = $v1["sotinchih"];
    $monhoc->save();
    return redirect()->route("lkmh")->with("thongbao", "Thêm môn học thành công");
}
public function get_suamonhoc($mamh)
{
    $monhoc = Monhoc::find($mamh);
    return view("buc07.suamonhoc", compact("monhoc"));
}
public function post_suamonhoc(Request $req, $mamh)
{
    $v1 = $req->validate([
        'tenmh' => 'required|min:7',
        'sotinchih' => 'required|numeric|min:3|max:10'
    ], [
        'tenmh.required' => 'Bạn chưa nhập tên môn học',
        'tenmh.min' => 'Tên môn học tối thiểu 7 ký tự',
        'sotinchih.required' => 'Chưa nhập số tín chỉ',
        'sotinchih.numeric' => 'phải nhập số',
        'sotinchih.min' => 'tối thiểu 3 tín chỉ',
        'sotinchih.max' => 'tối đa 10 tín chỉ'
    ]);
    $monhoc = Monhoc::find($mamh);
    $monhoc->tenmh = $v1["tenmh"];
    $monhoc->sotinchih = $v1["sotinchih"];
    $monhoc->save();
    return redirect()->route('lkmh')->with('thongbao', 'Cập nhật môn học thành công');
}
public function get_xoamonhoc($mamh)
{
    $monhoc = Monhoc::find($mamh);
    $monhoc->delete();
    return redirect()->route("lkmh")->with("thongbao", "Đã xóa môn học thành công");
}

```