**BÀI TẬP THỰC HÀNH SỐ 5**

**Học phần**: CSE485 - Công nghệ Web

**Mục tiêu:**

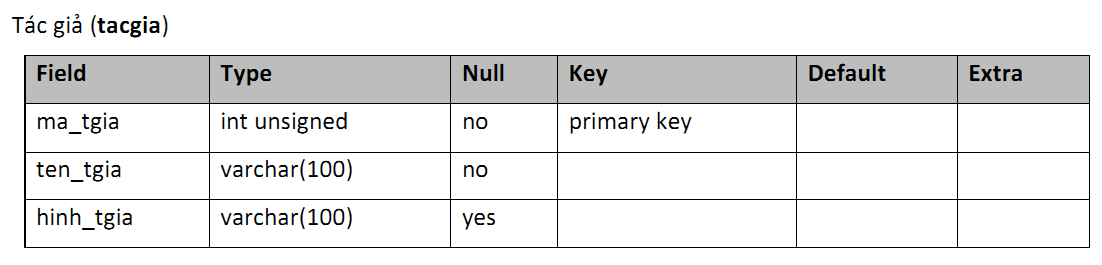
- Làm quen với cách tạo controller tùy chỉnh: không sử dụng **–resource**

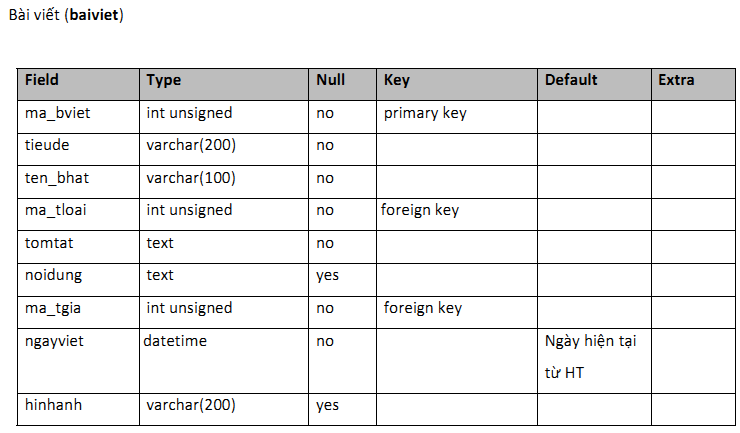
- Truy vấn dữ liệu theo các cách khác nhau trong Laravel: **Raw Data, Query Builder, Eloquent ORM**

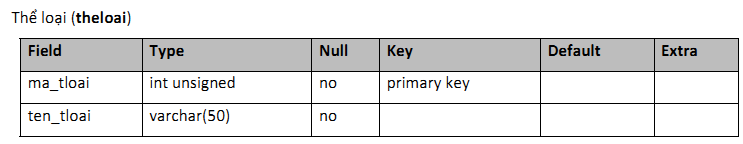
- Kiểm thử **URI/API** với **Postman**

**YÊU CẦU**

**1.** Tạo CSDL **Music** chứa và sinh ra dữ liệu giả cho 3 bảng như sau:

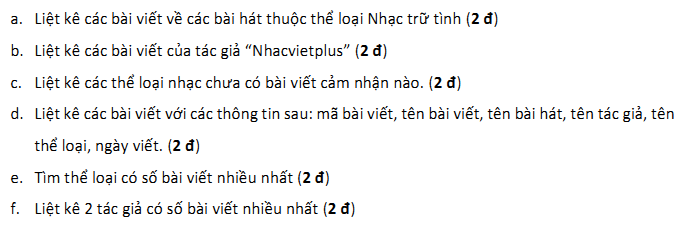


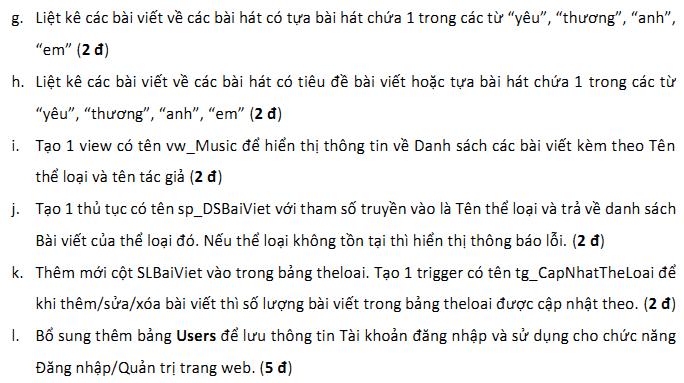




**2. Thực hiện các yêu cầu truy vấn trong Laravel với 3 cách khác nhau cho mỗi câu lệnh**

**(***Cách 1: Raw Data, Cách 2: Query Builder, Cách 3: Eloquent ORM***)**





3. Xây dựng các lớp Controller với các hàm phù hợp (ko sử dụng –resource) và định ngĩa các Route tương ứng.

4. Sử dụng Postman để kiểm thử các URI/API đã xây dựng.

**HƯỚNG DẪN**

**Bước 1**: Tạo Project

- Nếu ko cài đặt Laravel Installer:

composer create-project –prefer-dist laravel/laravel music-project

- Nếu máy đã có Laravel Installer:

laravel new music-project

**Bước 2**: Tạo CSDL. Giả sử đặt tên là music

**Bước 3**: Chỉnh sửa tệp .env trong dự án để khai báo kết nối CSDL

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=music

DB\_USERNAME=root

DB\_PASSWORD=abc

**Bước 4**: Tạo các tệp migrations (3 bảng/ tối thiểu 3 migrations)

- Tên bảng đề bài cho đang không đúng chuẩn để tạo tự động cùng Model > Tạo độc lập

php artisan make:migration create\_theloai\_table

php artisan make:migration create\_tacgia\_table

php artisan make:migration create\_baiviet\_table

**Bước 5**: Định nghĩa các tệp migrations

Schema::create('theloai', function (Blueprint $table) {

            $table->increments('ma\_tloai');

            $table->string('ten\_tloai',50);

            // $table->timestamps();

        });

Schema::create('tacgia', function (Blueprint $table) {

            $table->increments('ma\_tgia');

            $table->string('ten\_tgia',100);

            $table->string('hinh\_tgia',100);

            // $table->timestamps();

        });

Schema::create('baiviet', function (Blueprint $table) {

            $table->increments('ma\_bviet');

            $table->string('tieude',200);

            $table->string('ten\_bhat',100);

            $table->unsignedInteger('ma\_tloai');

            $table->text('tomtat');

            $table->text('noidung')->nullable();

            $table->unsignedInteger('ma\_tgia');

            $table->dateTime('ngayviet')->default(DB::raw('CURRENT\_TIMESTAMP'));

            $table->string('hinhanh',200)->nullable();

            $table->foreign('ma\_tloai')->references('ma\_tloai')->on('theloai')->onDelete('cascade');

            $table->foreign('ma\_tgia')->references('ma\_tgia')->on('tacgia')->onDelete('cascade');

            // $table->timestamps();

        });

Bước 6: Chạy lệnh migrate để thực thi script tạo Bảng

php artisan migrate

Bước 7: Tạo model

php artisan make:model **Category**

class Category extends Model

{

    protected $table = 'theloai';

    use HasFactory;

}

php artisan make:model **Author**

class Author extends Model

{

    protected $table='tacgia';

    use HasFactory;

}

php artisan make:model **Article**

class Article extends Model

{

    protected $table='baiviet';

    use HasFactory;

}

Bước 8: Gieo dữ liệu với Faker và Seeder

***Gợi ý***: Có thể sử dụng cách sinh dữ liệu theo cách thông thường, không cần Factory. id lấy theo biến đếm vòng lặp. Bảng chứa khóa ngoại sẽ sinh ngẫu nhiên giá trị trong khoảng của khóa chính.

- Gieo dữ liệu cho bảng theloai

php artisan make:seeder CategoryTableSeeder

use Faker\Factory as Faker;

use App\Models\Category;

$faker = Faker::create();

            for ($i = 0; $i < 10; $i++) {

                Category::create([

                    'ma\_tloai' => $i + 1,

                    'ten\_tloai' => $faker->word,

                ]);

            }

php artisan db:seed --class=CategoryTableSeeder

php artisan make:seeder AuthorTableSeeder

use Faker\Factory as Faker;

use App\Models\Author;

            $faker = Faker::create();

            for ($i = 0; $i < 10; $i++) {

                Author::create([

                    'ma\_tgia' => $i + 1,

                    'ten\_tgia' => $faker->word,

                    'hinh\_tgia' => $faker->word,

                ]);

php artisan db:seed --class=AuthorTableSeeder

php artisan make:seeder ArticleTableSeeder

use Faker\Factory as Faker;

use App\Models\Article;

use App\Models\Category;

use App\Models\Author;

$faker = Faker::create();

        $categories = Category::pluck('ma\_tloai');

        $authors = Author::pluck('ma\_tgia');

            for ($i = 0; $i < 10; $i++) {

                Article::create([

                    'ma\_bviet' => $i + 1,

                    'tieude' => $faker->sentence,

                    'ten\_bhat' => $faker->word,

                    'ma\_tloai' => $faker->randomElement($categories),

                    'tomtat' => $faker->word,

                    'noidung' => $faker->word,

                    // 'ma\_tgia' => $faker->numberBetween(1, 10),

                    'ma\_tgia' => $faker->randomElement($authors),

                    'ngayviet' => $faker->dateTime,

                    'hinhanh' => $faker->word,

                ]);

            }

php artisan db:seed --class=ArticleTableSeeder

B6 -> B8 TH4

php artisan make:controller CategoryController

Web.php

use App\Http\Controllers\CategoryController;

Route::resource('categories', CategoryController::class);

CategoryController

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Category;

class CategoryController extends Controller

{

    // public function index()

    // {

    // $categories = Category::all();

    // return view('categories.index', compact('categories'));

    // }

    public function index()

    {

        // Retrieve all the posts

        $categories = Category::orderBy('created\_at', 'desc')->paginate(10);

        // Pass the posts data to the view

        return view('categories.index', ['categories' => $categories]);

    }

    public function create()

    {

        // Just return the view

        return view('categories.create');

    }

    public function store(Request $request){

        // Validate the form data

        $validated = $request->validate([

            'ma\_tloai' => 'required|max:255',

            'ten\_tloai' => 'required|max:255',

        ]);

        // Create a new post instance and save it to the database

        $category = new Category;

        $category->title = $request->ma\_tloai;

        $category->body = $request->ten\_tloai;

        $category->save();

        // Redirect the user to the index view

        return redirect('/categories');

        }

    public function show($id)

    {

        // Retrieve the post by its id

        $category = Category::findOrFail($id);

        // Pass the post data to the view

        return view('categories.show', ['category' => $category]);

    }

}