



LARAVEL FRAMEWORK









BÀI 4: SỬ DỤNG MIGRATION & SEEDER

PHẦN 1: SỬ DỤNG MIGRATION






- ① Sử dụng Migration trong Laravel
- ① Sử dụng Seeder trong Laravel



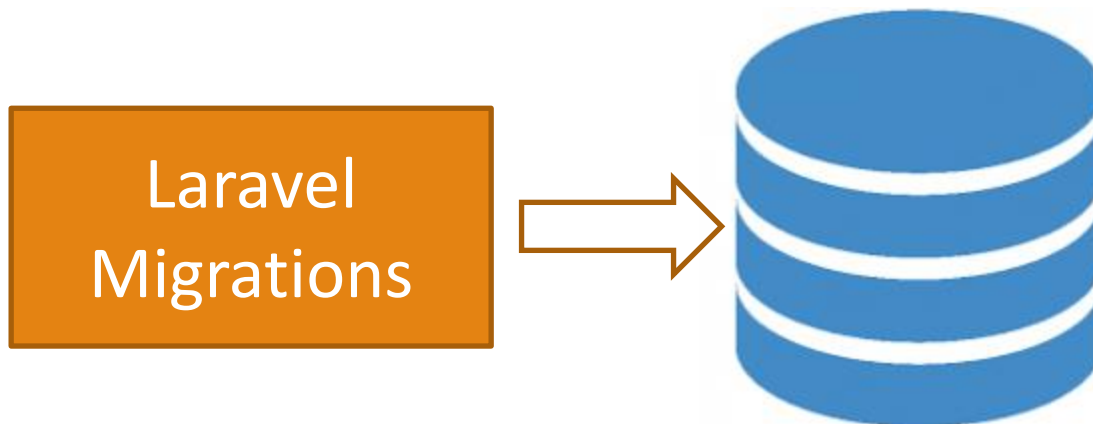
Phần I: Sử dụng Migration

-  Giới thiệu migration
-  Tạo migration
-  Migration Structure
-  Các lệnh có thể dùng trong migration
-  Thực thi, rolling back, Reset, Refresh migrations
-  Tạo table trong migration
-  Sửa. đổi tên, xóa table trong migration
-  Chỉ định chạy một migration

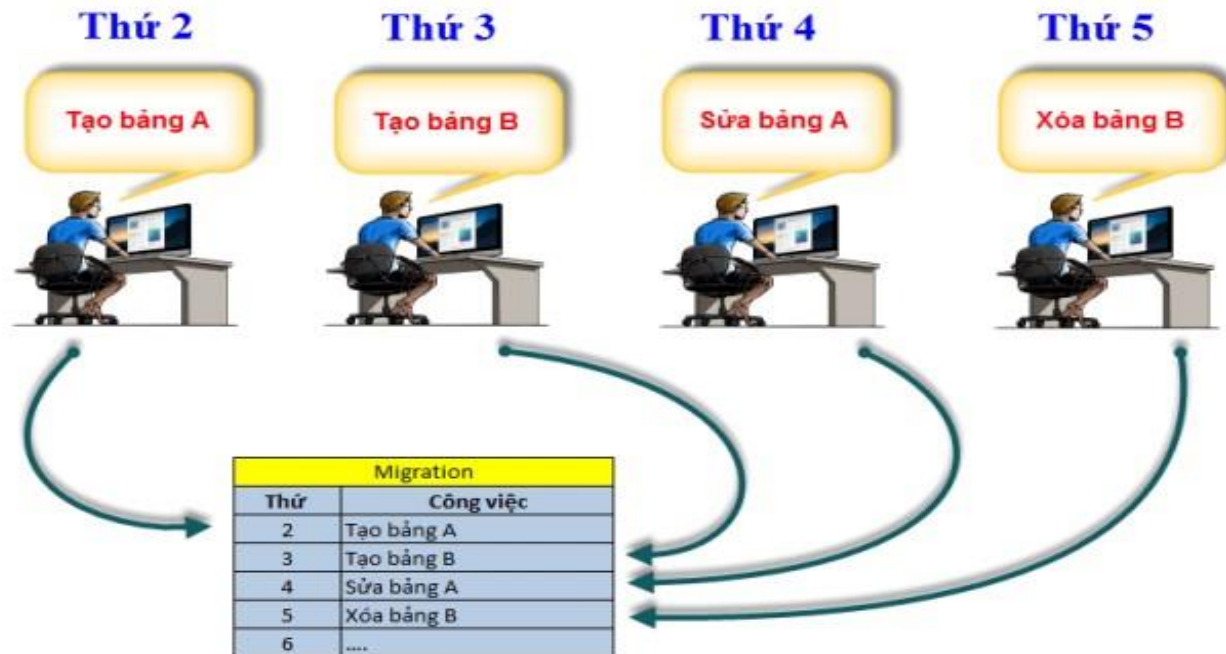
Phần II: Sử dụng Seeder

-  Giới thiệu seeder
-  Tạo mới một seeder
-  Seeder structure
-  Thực thi seeder
-  Thực thi seeder mặc định

- ❑ Migration là công cụ có sẵn trong Laravel giúp tạo và chỉnh sửa cấu trúc các table trong database rất dễ dàng.
- ❑ Mỗi migration là 1 file lưu trong folder **database/migrations**
- ❑ Mỗi migration được tạo ra để **tạo** hoặc **chỉnh cấu trúc** một table nào đó trong database.



- ❑ Migration khi tạo xong có thể chạy để thực thi, chạy để rollback (hủy bỏ hành động) lại migration cũng được.
- ❑ Migration rất cần thiết trong quá trình phát triển dự án, do nhu cầu điều chỉnh nhiều lần database, đặc biệt trong team nhiều người và làm việc xa nhau.

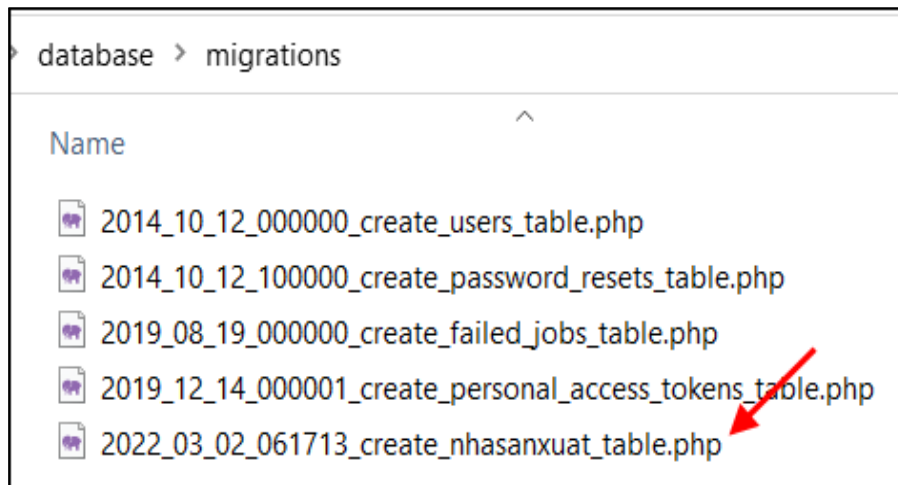


- ❑ Để tạo migration cho mục đích tạo table (nhasanxuat), thực hiện như sau:

php artisan make:migration taotableNsx --create=nhasanxuat
hoặc

php artisan make:migration create_nhasanxuat_table

- ❑ Vào folder database/migrations sẽ thấy migration mới tạo

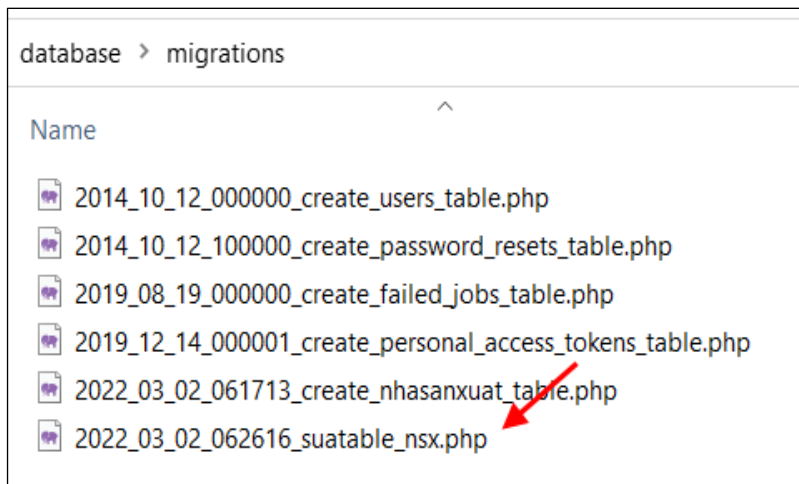


```
return new class extends Migration {  
    public function up() {  
        Schema::create('nhasanxuat', function($table) {  
            $table->id();  
            $table->timestamps();  
        });  
    }  
    public function down() {  
        Schema::dropIfExists('nhasanxuat');  
    }  
};
```

- ❑ Để tạo migration dùng cho mục đích chỉnh cấu trúc table (nhasansuat), sử dụng lệnh như sau:

php artisan make:migration suatableNsx --table=nhasansuat

- ❑ Vào folder database/migrations, sẽ thấy migration mới tạo



```
return new class extends Migration{
    public function up()    {
        Schema::table('nhasansuat', function (Blueprint $table) {
            //
        });
    }
    public function down() {
        Schema::table('nhasansuat', function (Blueprint $table) {
            //
        });
    }
};
```

- ❑ Mỗi file migration được Laravel tạo sẵn hai hàm: **up** và **down**. Hàm **up** dùng để tạo mới tables/columns. Hàm **down** thực hiện ngược lại với up (xóa)

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration {
    public function up() {
        Schema::create('nhasanxuat', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('nhasanxuat');
    }
};
```


- ❑ Bên trong 2 hàm up ; down , Bạn có thể code để tạo/ sửa table với các field và thuộc tính cần thiết.

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration {
    public function up() {
        Schema::create('nhasanxuat', function (Blueprint $table) {
            $table->id();
            $table->string('ten');
            $table->string('logo');
            $table->boolean('status');
            $table->timestamps();
            $table->charset = 'utf8mb4';
            $table->collation = 'utf8mb4_unicode_ci';
        });
    }
    public function down() {
        Schema::dropIfExists('nhasanxuat');
    }
};
```

Một số lệnh thường dùng trong migration để khai báo các field và thuộc tính cho table:

- ❑ **\$table->increments('id');** → field id tăng tự động, unsigned integer
- ❑ **\$table->id();** → field id tăng tự động, big Integer
- ❑ **\$table->integer('soLuong');** → field kiểu integer, tên soLuong
- ❑ **\$table->string('tenSP', 100);** → field kiểu varchar, độ dài 100
- ❑ **\$table->boolean('anHien');** → field kiểu Boolean
- ❑ **\$table->double('diemTB', 8, 2);** → field kiểu số thực
- ❑ **\$table->dateTime('thoiDiemMuaHang');** → field kiểu ngày giờ
- ❑ **\$table->charset = 'utf8mb4';** → Khai báo charset cho table
- ❑ **\$table->collation = 'utf8mb4_unicode_ci';** → Khai báo collation cho table
- ❑ **\$table->timestamps();** → Tạo 2 field created_at và updated_at
- ❑ Tham khảo thêm tại:

<https://laravel.com/docs/migrations#creating-columns>

- ❑ Để chạy tất cả các file migrations chưa được chạy, bạn mở command line, chuyển vào folder project rồi gõ lệnh **php artisan migrate**

```
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (35.38ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (49.18ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (54.42ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (70.04ms)
Migrating: 2022_03_02_061713_create_nhasanxuat_table
Migrated: 2022_03_02_061713_create_nhasanxuat_table (16.20ms)
Migrating: 2022_03_02_062616_suatable_nsx
Migrated: 2022_03_02_062616_suatable_nsx (0.26ms)
```

- ❑ Để rollback lại thao tác migration cuối cùng, bạn dùng lệnh **php artisan migrate:rollback**
- ❑ Lệnh này sẽ rollback lại nhóm các migration được chạy lần gần nhất.

```
Rolling back: 2022_03_02_062616_suatable_nsx
Rolled back: 2022_03_02_062616_suatable_nsx (11.47ms)
Rolling back: 2022_03_02_061713_create_nhasanxuat_table
Rolled back: 2022_03_02_061713_create_nhasanxuat_table (10.92ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (12.14ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (11.32ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (11.49ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (11.57ms)
```

- ❑ Lệnh **php artisan migrate:reset** sẽ rollback lại tất cả migration của ứng dụng.

```
E:\PHP3\myProject>php artisan migrate:reset  
Nothing to rollback.  
E:\PHP3\myProject>
```

- ❑ Lệnh **php artisan migrate:refresh** sẽ rollback lại toàn bộ migration và rồi chạy lệnh **migrate**.

```
Nothing to rollback.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table (39.96ms)  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table (45.69ms)  
Migrating: 2019_08_19_000000_create_failed_jobs_table  
Migrated: 2019_08_19_000000_create_failed_jobs_table (37.02ms)  
Migrating: 2019_12_14_000001_create_personal_access_tokens_table  
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (53.46ms)  
Migrating: 2022_03_02_061713_create_nhasanxuat_table  
Migrated: 2022_03_02_061713_create_nhasanxuat_table (19.88ms)  
Migrating: 2022_03_02_062616_suatable_nsx  
Migrated: 2022_03_02_062616_suatable_nsx (0.19ms)
```

- ❑ Trong migration, lệnh **Schema::create** dùng để tạo table mới. Bạn viết nó trong hàm up và cú pháp như sau

```
Schema::create('nhasanxuat', function(Blueprint $table){  
    $table->id();  
    $table->string('ten');  
    $table->string('logo');  
    $table->boolean('status');  
    $table->timestamps();  
    //...  
    $table->charset = 'utf8mb4';  
    $table->collation = 'utf8mb4_unicode_ci';  
    $table->engine = 'InnoDB';  
});
```

- ❑ Tham số đầu tiên của create là tên table muốn tạo.
- ❑ Tham số thứ hai là hàm khai báo các field và các thuộc tính của table.

- ❑ Trong migration, lệnh **Schema::table** dùng để sửa cấu trúc table. Lệnh có cú pháp như sau:

```
Schema::table('nhasanxuat', function (Blueprint $table) {  
    $table->boolean('status');  
    $table->engine = 'MyISAM';  
});
```

- ❑ Tham số đầu tiên là tên table cần sửa, tham số thứ hai là nơi khai báo các field và các thuộc tính cần sửa của table
- ❑ Để test, tạo 1 migration update table và viết lệnh trong hàm up. Xong thì thực thi với lệnh **php artisan migrate**

❑ Lệnh đổi tên table

```
Schema::rename("nhasanxuat", "nsx");//from=>to
```

❑ Lệnh xóa table

```
Schema::drop('nsx');  
Schema::dropIfExists('nsx');
```

❑ Để test, tạo 1 migration update table và viết lệnh trong hàm up. Xong thì thực thi với lệnh **php artisan migrate**

- ❑ Để chạy một migration cụ thể, thực hiện lệnh
php artisan migrate --path=database\migrations\tênFile.php
- ❑ Vài lưu ý:
 - ❖ Bạn muốn chạy lại 1 migration vừa mới chạy, có thể vào table **migrations** trong database và xóa dòng tương ứng với migration cần chạy.
 - ❖ Với 1 table cụ thể đã có, bạn muốn thêm 1 field **created_at** và **updated_at** thì tạo migration mới và code trong hàm up với lệnh timestamps() như sau

```
public function up() {  
    Schema::table('products', function ( Blueprint $table) {  
        $table->timestamps();  
    });  
}
```



DEMO

- Demo tạo, thực thi, rollback migrat



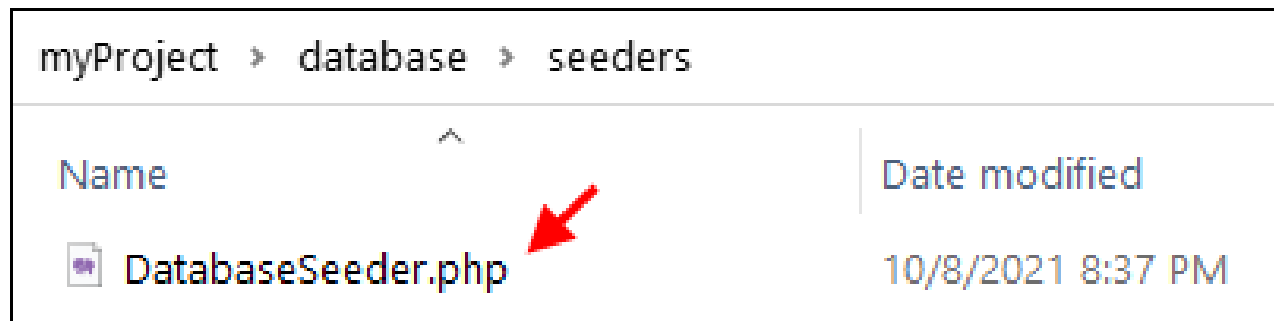


LARAVEL FRAMEWORK

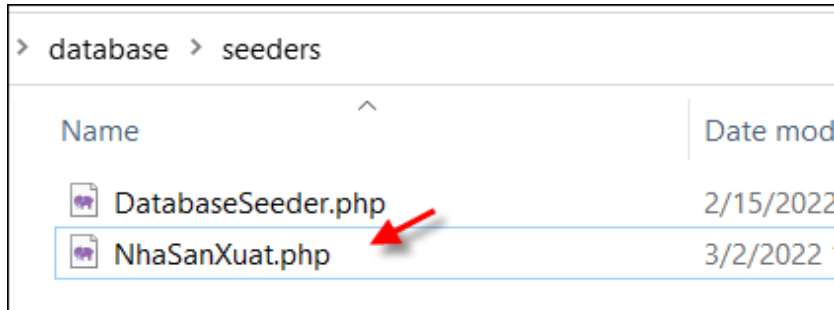
BÀI 4: SỬ DỤNG MIGRATION & SEEDER

PHẦN 2: SỬ DỤNG SEEDER

- ❑ Seeder là công cụ trong Laravel giúp tạo dữ liệu trong các table một cách nhanh chóng và dễ dàng.
- ❑ Mỗi seeder là 1 file, lưu trong folder **database/seiders**
- ❑ Trong folder **seeders**, Laravel đã tạo sẵn một seeder với tên **DatabaseSeeder** để bạn dùng, nhưng có thể tạo seeder mới.
- ❑ Mỗi seeder được tạo ra là để chèn, sửa, xóa dữ liệu trong các table của database.
- ❑ Seeder rất cần thiết trong quá trình phát triển dự án, do nhu cầu cần có nhiều dữ liệu để trình bày trong trang web.



- ❑ Để tạo 1 seeder mới, dùng lệnh :
php artisan make:seeder NhaSanXuat
- ❑ Vào folder database/seeder , sẽ thấy file mới tạo



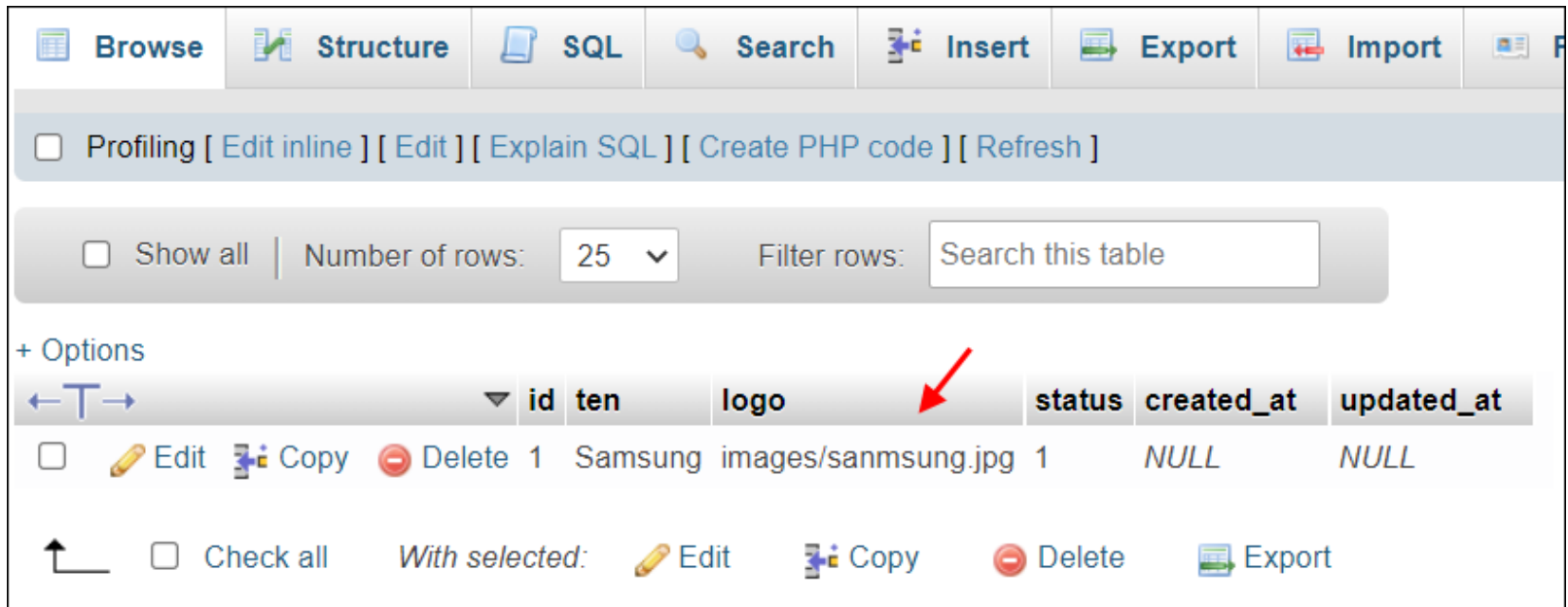
> database > seeders	
Name	Date modified
DatabaseSeeder.php	2/15/2022
NhaSanXuat.php	3/2/2022

```
<?php
// seeders/NhaSanXuat.php
namespace Database\Seeders;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
class NhaSanXuat extends Seeder {
    public function run() {
        // ← Code trong seeder viết ở đây
    }
}
```

- ❑ Mỗi file seeder được Laravel tạo sẵn một hàm: **run**. Hàm này là nơi bạn viết code để thao tác trên dữ liệu trong các table (thêm, sửa, xóa)
- ❑ Code tương tác db trong hàm run có thể dùng Query Builder hoặc Eloquent trong Laravel

```
<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
class NhaSanXuat extends Seeder {
    public function run() {
        \DB::table('nhasanxuat')->insert([
            'ten' => 'Samsung',
            'logo' => 'images/sanmsung.jpg',
            'status' => 1
        ]);
    }
}
```

- ❑ Để chạy seeder đã tạo, chạy lệnh sau trong folder project
php artisan db:seed --class=NhaSanXuat
- ❑ Vào database sẽ thấy tác dụng

























The screenshot shows a database management interface with a table containing one row. A red arrow points to the 'logo' column.

	id	ten	logo	status	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Samsung	images/sanmsung.jpg	1	NULL	NULL

At the bottom of the interface, there are options to 'Check all', 'With selected:', 'Edit', 'Copy', 'Delete', and 'Export'.

- ❑ Trong seeder có thể để chèn nhiều dòng dữ liệu cùng lúc

```
<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
class NhaSanXuat extends Seeder {
    public function run() {
        \DB::table('nhasanxuat')->insert([
            ['ten' => 'HTC', 'logo'=>'', 'status' => 1],
            ['ten' => 'Acer', 'logo'=>'', 'status' => 0],
            ['ten' => 'Lonovo', 'logo'=>'', 'status' => 2]
        ]);
    }
}
```

 Browse		 Structure		 SQL		 Search		 Insert		 Export		 Import			
						id	ten	logo		status	created_at	updated_at			
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Samsung	images/sanmsung.jpg		1	NULL	NULL					
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	HTC			1	NULL	NULL					
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Acer			0	NULL	NULL					
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	Lonovo			2	NULL	NULL					

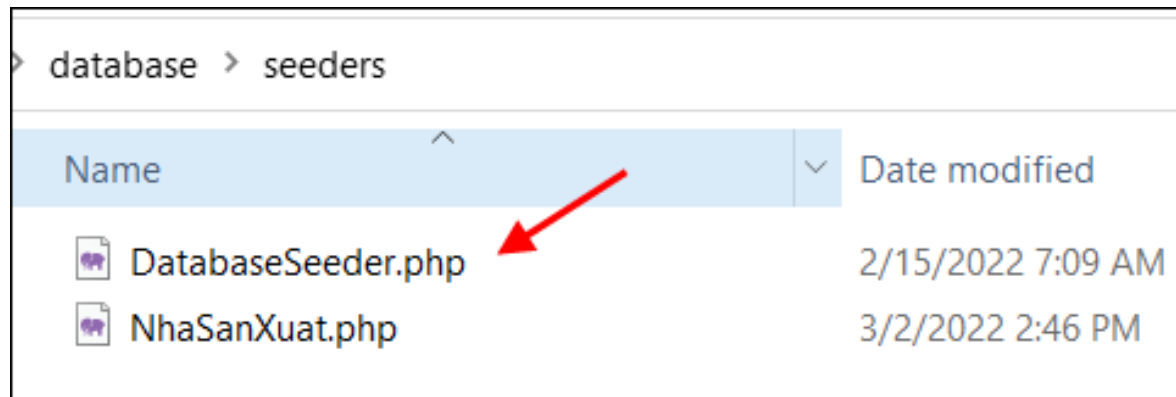
- ❑ Trong Seeder có thể phát sinh dữ liệu ngẫu nhiên

```
<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
class NhaSanXuat extends Seeder {
    public function run() {
        for ($i=0; $i<50; $i++) {
            \DB::table('nhasanxuat')->insert([
                'ten' => 'Dell ' . rand(0,10),
                'logo'=>'',
                'status'=> rand(0,2)
            ]);
        } //for
    }
}
```

id	ten	logo	status	created_at	updated_at
4	Lonovo		2	NULL	NULL
5	Dell 3		1	NULL	NULL
6	Dell 9		0	NULL	NULL
7	Dell 10		0	NULL	NULL
8	Dell 3		2	NULL	NULL
9	Dell 9		2	NULL	NULL
10	Dell 8		2	NULL	NULL

- ❑ Trong ngữ cảnh đơn giản, bạn có thể không cần tạo seeder mới mà dùng luôn seeder có sẵn là DatabaseSeeder
- ❑ Để chạy seeder mặc định DatabaseSeeder của Laravel, bạn chỉ cần gõ lệnh sau trong command line:

php artisan db:seed













DEMO






- Tạo và sử dụng master page
- Xây dựng Blade dùng Control structure



Phần I: Sử dụng Migration

-  Giới thiệu migration
-  Tạo migration
-  Migration Structure
-  Các lệnh có thể dùng trong migration
-  Thực thi, rolling back, Reset, Refresh migrations
-  Tạo table trong migration
-  Sửa. đổi tên, xóa table trong migration
-  Chỉ định chạy một migration

Phần II: Sử dụng Seeder

-  Giới thiệu seeder
-  Tạo mới một seeder
-  Seeder structure
-  Thực thi seeder
-  Thực thi seeder mặc định





Cảm ơn

- ❑ <https://laravel.com/docs/migrations>
- ❑ <https://laravel.com/docs/migrations#creating-columns>