

Lab 4: Database: Migrations



MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Cấu hình lệnh thực thi Migration
- ✓ Tạo Database, table, seeding
- ✓ Laravel rollback

BÀI 1 (2 ĐIỂM)

Tạo Database, table và cấu hình connection cho Artisan và Laravel, sử dụng lệnh cmd artisan

HƯỚNG DẪN:

- ✓ Mở PHPMysqlAdmin để tạo mới database có tên “larashop”:

```
1 | CREATE DATABASE `larashop`;
```

- ✓ Cấu hình các tham số cho Laravel bên trong /config/database.php

```
1 | 'mysql' => [  
2 |     'driver'      => 'mysql',  
3 |     'host'        => env('DB_HOST', 'localhost'),  
4 |     'database'    => env('DB_DATABASE', 'forge'),  
5 |     'username'    => env('DB_USERNAME', 'forge'),  
6 |     'password'    => env('DB_PASSWORD', ''),  
7 |     'charset'     => 'utf8',  
8 |     'collation'   => 'utf8_unicode_ci',  
9 |     'prefix'      => '',  
10 |    'strict'      => false,  
11 | ],
```

Sinh cần thay đổi phù hợp với database thực tế của sinh viên tạo về tên

```

1 | 'database' => env('DB_DATABASE', 'larashop'),
2 | 'username' => env('DB_USERNAME', 'root'),
3 | 'password' => env('DB_PASSWORD', 'melody'),

```

- ✓ Cấu hình Database connection cho Atisan bên trong file “.env”

```

1 | APP_ENV=local
2 | APP_DEBUG=true
3 | APP_KEY=aqk5XHULL8TZ8t6pXE43o7MBSFchfgy2
4 |
5 | DB_HOST=localhost
6 | DB_DATABASE=homestead
7 | DB_USERNAME=homestead
8 | DB_PASSWORD=secret
9 |
10 | CACHE_DRIVER=file
11 | SESSION_DRIVER=file
12 | QUEUE_DRIVER=sync
13 |
14 | MAIL_DRIVER=smtp
15 | MAIL_HOST=mailtrap.io
16 | MAIL_PORT=2525
17 | MAIL_USERNAME=null
18 | MAIL_PASSWORD=null
19 | MAIL_ENCRYPTION=null

```

- ✓ Sinh viên chú ý thay đổi thông tin “database, username and password” sao cho phù hợp, ví dụ trong trường hợp sử dụng cơ sở dữ liệu “larashop” với pass là “melody” và user là “root” thì sẽ cấu hình như sau:

```

1 | DB_HOST=localhost
2 | DB_DATABASE=larashop
3 | DB_USERNAME=root
4 | DB_PASSWORD=melody

```

- ✓ Sau bước cấu hình, sinh viên tạo table bằng các lệnh artisan migration. Khi dùng migration thì Laravel mặc định lưu trong thư mục

/database/migrations.

- ✓ Chạy Cmd để duy chuyển vào thư mục project larashop

```
1 | cd C:\xampp\htdocs\larashop
```

- ✓ Tạo Migration table

```
1 | php artisan migrate:install
```

- ✓ Chạy thành công nhận được thông báo

```
1 | Migration table created successfully.
```

- ✓ Chạy lệnh tạo file migration

```
1 | php artisan make:migration create_drinks_table
```

- `php artisan make:migration` Thông qua artisan để thực thi phương thức của migration
- `createdrinkstable` Chỉ định tên cho file migration được tạo. Lưu ý là cột timestamp cũng sẽ được thêm tự động vào file migration

- ✓ Thông báo tạo thành công

```
1 | Created Migration: 2015_08_27_072434_create_drinks_table
```

BÀI 2 (2 ĐIỂM):

Tìm hiểu cấu trúc file migration và tạo table từ bên trong file

HƯỚNG DẪN:

- ✓ Trong bài 1 sau khi sinh viên tạo xong file migration `20150827072434createdrinkstable`, file này được lưu trữ trong `/database/migrations`. Sinh viên mở `/database/migrations/20150827072434createdrinkstable.php` và xem qua

cấu trúc file này:

```
1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class CreateDrinksTable extends Migration
7  {
8      public function up()
9      {
10         //
11     }
12
13     public function down()
14     {
15         //
16     }
17 }
```

- `class CreateDrinksTable extends Migration` Tạo lớp CreateDrinksTable kế thừa lớp Migration
 - `public function up()` Định nghĩa hàm thực thi khi chạy migration
 - `public function down()` Định nghĩa hàm thực thi khi migration rollback
- ✓ Định nghĩa Table từ file migration
"/database/migrations/20150827072434createdrinkstable.php" đã tạo trong bài 1

```

1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class CreateDrinksTable extends Migration {
7
8      /**
9       * Run the migrations.
10      *
11      * @return void
12      */
13     public function up() {
14         Schema::create('drinks', function (Blueprint $table) {
15             $table->increments('id');
16             $table->string('name',75)->unique();
17             $table->text('comments')->nullable();
18             $table->integer('rating');
19             $table->date('brew_date');
20             $table->timestamps();        });
21     }

```

```

22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down() {
29         Schema::drop('drinks');
30     }
31 }

```

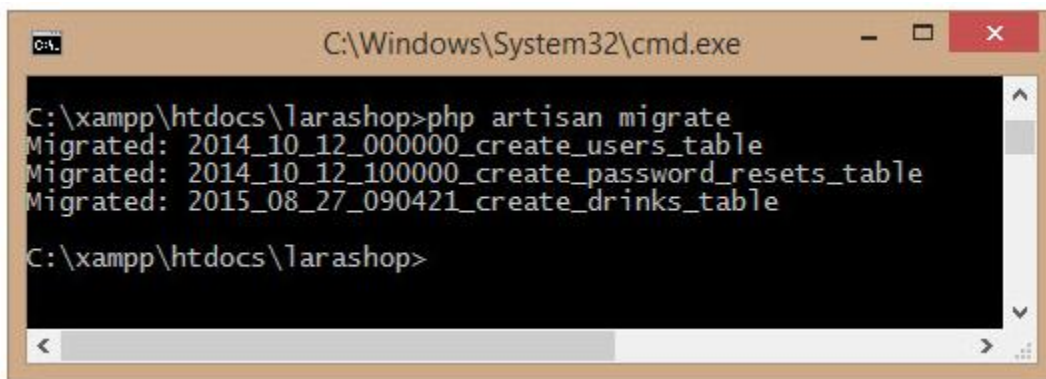
- `Schema::create('drinks', function (Blueprint $table) {..})` Gọi hàm khởi tạo từ lớp Schema. Tham số `$table` dùng định nghĩa cấu trúc cho database.
- `$table->increments('id');` Tạo field Id có giá trị tự tăng
- `$table->string('name',75)->unique();` Tạo field Name kiểu chuỗi có độ dài 75 và `->unique()` xác định tính duy nhất
- `$table->text('comments')->nullable();` Tạo text field. `->nullable()` Cho phép field này nhận giá trị null

FPT POLYTECHNIC

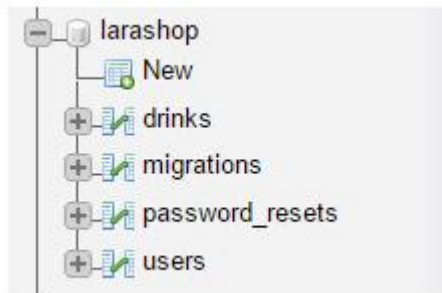
- `$table->integer('rating');` tạo field rating kiểu interger
- `$table->date('brew_date');` tạo field brew_date kiểu ngày tháng
- `$table->timestamps();` do hệ thống tạo ra tự động lưu lại thời gian giữa lúc tạo và chỉnh sửa.

✓ Để thực thi các lệnh trên, sinh viên quay lại chạy lên CMD

```
1 | php artisan migrate
```



✓ Sinh viên mở PHPMyAdmin kiểm tra cơ sở dữ liệu Larashop vừa được tạo



- ✓ Laravel xây dựng cơ chế RollBack giúp khôi phục tình trạng như trước lúc thực hiện migration. Ví dụ chúng ta có thể rollback về như trước lúc tạo table
- ✓ Gõ lệnh rollback trên cmd

```
1 | php artisan migrate:rollback
```

FPT POLYTECHNIC

- ✓ Kết quả khi chạy rollback

```
1 | Rolled back:2018_10_12_000000_create_users_table.php
2 | Rolled back:2018_10_12_100000_create_password_resets_table.php
3 | Rolled back:2018_08_27_090421_create_drinks_table.php
```

- ✓ Xác nhận rollback bằng lệnh

```
1 | php artisan migrate
```

BÀI 3 (2 ĐIỂM)

- ✓ Thực hiện thêm dữ liệu vào table Employees, chạy lệnh tạo file Employee migration

```
1 | php artisan make:migration employees
```

- ✓ Mở file /database/migrations/xxxxxxxxx_employees.php với xxxxxxxxxx là timestamp do migration gắn vào cuối file

```
1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class Employees extends Migration
7  {
8      /**
9       * Run the migrations.
10      *
11      * @return void
12      */
13     public function up()
14     {
15         Schema::create('employees', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('name');
18             $table->string('email')->unique();
19             $table->string('contact_number');
20             $table->timestamps();
21         });
```



```

22
23     $faker = Faker\Factory::create();
24
25     $limit = 33;
26
27     for ($i = 0; $i < $limit; $i++) {
28         DB::table('employees')->insert([ //,
29             'name' => $faker->name,
30             'email' => $faker->unique()->email,
31             'contact_number' => $faker->phoneNumber,
32         ]);
33     }
34 }
35
36 /**
37  * Reverse the migrations.
38  *
39  * @return void
40  */
41 public function down()
42 {
43     Schema::drop('employees');
44 }
45 }

```

- `$faker = Faker\Factory::create();` tạo instance từ Faker factory
- `$limit = 33;` Xác định số record thêm vào table là 33.
- `for ($i = 0; $i < $limit; $i++) { DB::table('employees')->insert([...]); }` Chạy vòng lặp 33 lần để thêm record.

✓ Xác nhận chạy lệnh migration thêm dữ liệu

```
1 | php artisan migration
```

```
1 | Migrated: xxxxxxxx_employees.php
```

✓ Để kiểm tra kết quả, sinh viên vào MySql chạy lệnh

```
1 | SELECT * FROM employees;
```

- ✓ Laravel cũng hỗ trợ thêm columns hoặc xóa columns dễ dàng. Ví dụ thêm column “gender” sau column “contact_number” trong table employee, chạy lệnh:

```
1 | php artisan make:migration add_gender_to_employees --table=employees
```

- ✓ Update lại code bên trong
/database/migration/xxxxxxxxxxxxxxx_add_gender_to_employees.php

```
1 | <?php
2 |
3 | use Illuminate\Database\Schema\Blueprint;
4 | use Illuminate\Database\Migrations\Migration;
5 |
6 | class AddGenderToEmployees extends Migration
7 | {
8 |     /**
9 |      * Run the migrations.
10 |      *
11 |      * @return void
12 |      */
13 |     public function up()
14 |     {
15 |         Schema::table('employees', function (Blueprint $table) {
16 |             $table->string('gender')->after('contact_number');
17 |         });
18 |     }
19 | }
```

```

19
20     /**
21     * Reverse the migrations.
22     *
23     * @return void
24     */
25     public function down()
26     {
27         Schema::table('employees', function (Blueprint $table) {
28             $table->dropColumn('gender');
29         });
30     }
31 }

```

- `public function up(){...}` uses `Schema::table('employees'...)` Thêm mới column “gender”

BÀI 4 (2 ĐIỂM) : Thiết lập khóa ngoại trong migration

- ✓ Tạo thêm table “depts” để quản lý các employee. Thêm khóa ngoại cho dept_id, lần lượt làm các bước sau:

```
1 | php artisan make:migration depts
```

```

1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class Depts extends Migration
7  {
8      /**
9       * Run the migrations.
10      *
11      * @return void
12      */
13     public function up()
14     {
15         Schema::create('depts', function (Blueprint $table) {
16             $table->increments('id');
17             $table->string('name');
18             $table->timestamps();
19         });
20     }

```

```

21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::drop('depts');
30     }
31 }

```

```
1 | php artisan migrate
```

- ✓ Lưu ý cần đảm bảo khóa chính và khóa ngoại ở cả 2 table phải cùng kiểu dữ liệu và cùng độ dài
- ✓ Thêm dept_id vào table employees

```
1 | php artisan make:migration add_dept_id_in_employees --table=employees
```

- ✓ Tiến hành tạo ràng buộc khóa chính khóa ngoại

```
1  <?php
2
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5
6  class AddDeptIdInEmployees extends Migration
7  {
8      /**
9       * Run the migrations.
10      *
11      * @return void
12      */
13     public function up()
14     {
15         Schema::table('employees', function (Blueprint $table) {
16             $table->unsignedInteger('dept_id')->after('gender');
17             $table->foreign('dept_id')
18                 ->references('id')->on('depts')
19                 ->onDelete('cascade');
20         });
21     }
```

```
22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down()
29     {
30         Schema::table('employees', function (Blueprint $table) {
31             $table->dropColumn('dept_id');
32         });
33     }
34 }
```

- ✓ Chạy thực thi migration và vào MySql kiểm tra kết quả

```
1 | php artisan migrate
```

BÀI 5 (2 ĐIỂM) : Giảng viên cho thêm