Fpt University

FPT POLYTECHNIC

LARAVEL FRAMEWORK

BÀI 4: DATABASE: MIGRATIONS

www.poly.edu.vn

Mục tiêu

- Nắm vững cấu trúc,trường hợp sử dụng Migration.
- Tạo và sử dụng Migration

NỘI DUNG

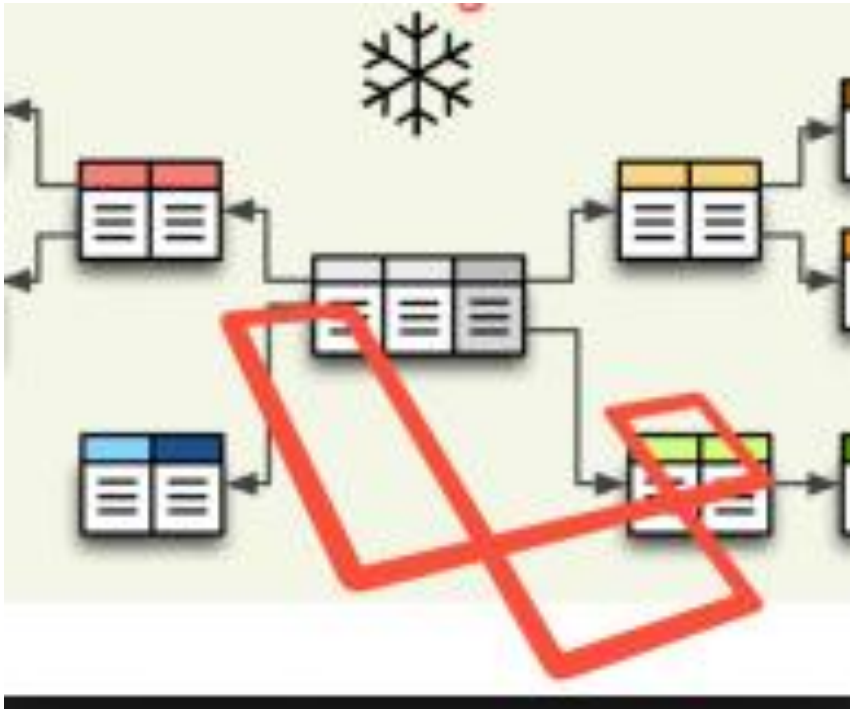
- Phần I: Giới thiệu về Migration
  - Làm quen Migration
  - Các yêu cầu cần có khi chạy Migration
  - Cấu trúc và thực thi Migration
- Phần II: Tạo và làm việc với Migration
  - Tables
  - Columns
  - Foreign Key Constraints

TỔNG QUAN MIGRATION

- Migration được coi như là version control cho database, cho phép team có thể dễ dàng thay đổi và chia sẻ schema của database trong chương trình với nhau.

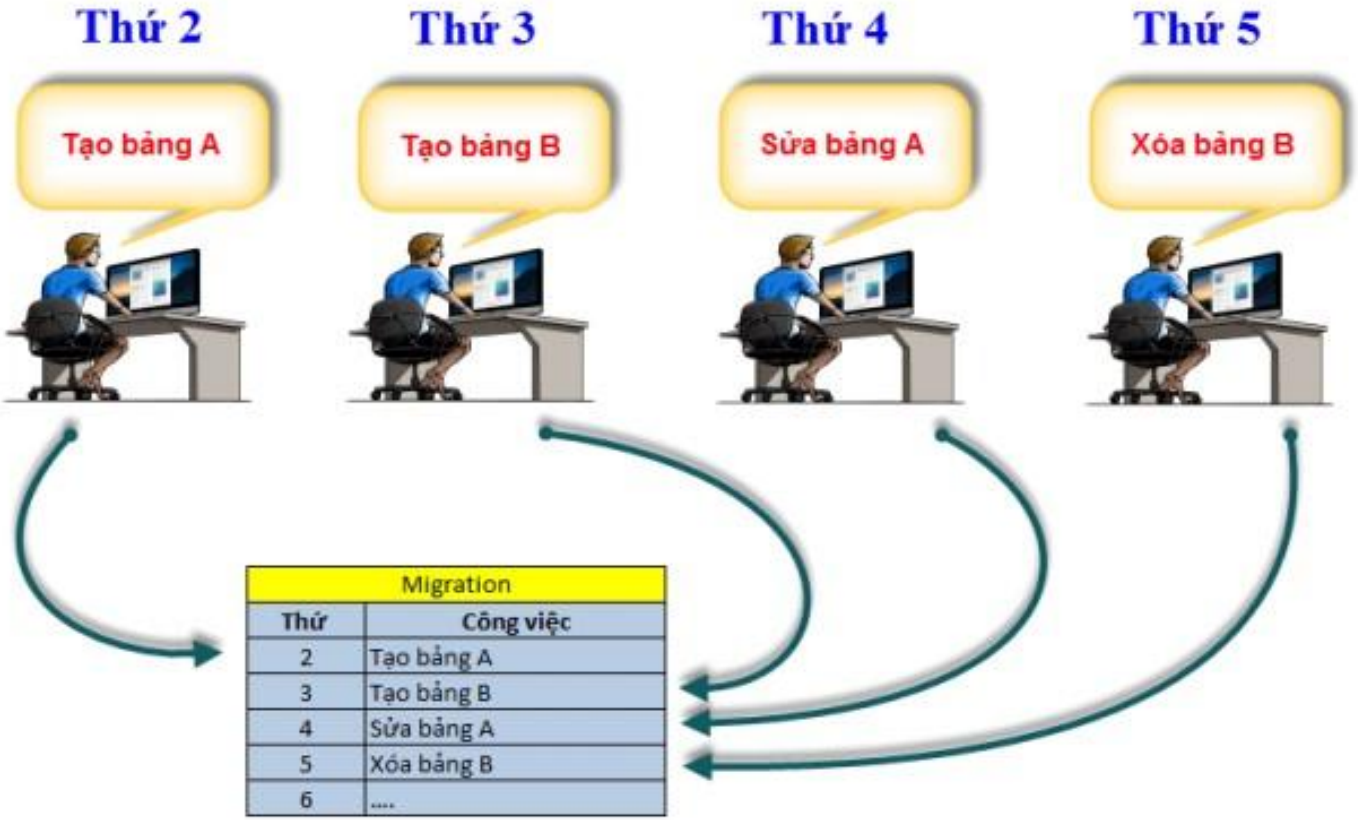
TỔNG QUAN MIGRATION

- Migration cơ bản được sử dụng cùng với schema builder để dễ dàng xây dựng cấu trúc cho database schema.



TỔNG QUAN MIGRATION

- Cho phép backup, reset, rollback lại cơ sở dữ liệu hay là tạo một bộ cơ sở dữ liệu mẫu để chúng ta làm việc với database một cách dễ dàng hơn



CẤU HÌNH CHẠY MIGRATION

- Tạo database từ PhpMyAdmin

CẤU HÌNH CHẠY MIGRATION

- Cấu hình file /config/database.php
  - Loại cơ sở dữ liệu mặc đinh là mysql, có thể sử dụng sqlite, pgsql…

```
/*
|--------------------------------------------------------------------------
| Default Database Connection Name
|--------------------------------------------------------------------------
|
| Here you may specify which of the database connections below you wish
| to use as your default connection for all database work. Of course
| you may use many connections at once using the Database library.
|
*/

'default' => 'mysql',

/*
```

CẤU HÌNH CHẠY MIGRATION

- Cấu hình file /config/database.php
  - Các thông tin cấu hình cho database

```
'mysql' => [
    'driver'    => 'mysql',
    'host'      => env('DB_HOST', 'localhost'),
    'database'  => env('DB_DATABASE', 'forge'),
    'username'  => env('DB_USERNAME', 'forge'),
    'password'  => env('DB_PASSWORD', ''),
    'charset'   => 'utf8',
    'collation' => 'utf8_unicode_ci',
    'prefix'    => '',
    'strict'    => false,
],
```

CẤU HÌNH CHẠY MIGRATION

- Cấu hình file .env

```
APP_ENV=local
APP_DEBUG=true
APP_KEY=aqk5XHULL8TZ8t6pXE43o7MBSFchfgy2

DB_HOST=localhost
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

MAIL_DRIVER=smtp
MAIL_HOST=mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

CẤU HÌNH CHẠY MIGRATION

- Kiểm tra kết nối Migration
  - Sử dụng Cmd tới thư mục chứa project

```
PS W:\xampp\htdocs\myproject>
```

  - Chạy lệnh php artisan migrate

```
PS W:\xampp\htdocs\myproject> php artisan migrate
```

  - Nếu thành công

```
C:\wamp\www\laravel-5>php artisan migrate
Migration table created successfully.
Migrated: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_100000_create_password_resets_table
```

GENERATING MIGRATIONS

- Sử dụng lệnh : php artisan make:migration create_users_table
- File migration mới sẽ được đặt trong thư mục database/migrations. Mỗi file migration được đặt tên bao gồm timestamp để xác định thứ tự các migration với nhau.

```
php artisan make:migration create_drinks_table
```

```
Created Migration: 2015_08_27_072434_create_drinks_table
```

MIGRATION STRUCTURE

- Vào thư mục database/migrations sẽ thấy file 2018_02_01_create_Name_table.php với Name là tên table vừa tạo
- Một migration class chứa hai hàm cơ bản là: up và down. Hàm up được dùng để tạo table, cột hay index mới vào trong database, trong khi hàm down đơn giản chỉ dùng để rollback ngược lại những thao tác ở hàm up.

MIGRATION STRUCTURE

```php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateFlightsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('flights', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('airline');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('flights');
    }
```

RUNNING MIGRATIONS

- Thực thi các migration

```
php artisan migrate
```

- Thực thi migration không cần xác thực

```
php artisan migrate --force
```

ROLLING BACK MIGRATIONS

- Để rollback lại thao tác migration cuối cùng, bạn có thể sử dụng câu lệnh rollback. Chú ý là việc rollback này sẽ thực hiện lại "nhóm" những migration được chạy lần gần nhất, có thể là một hay nhiều files

```
php artisan migrate:rollback
```

- Có thể rollback migration tại thời điểm cố định

```
php artisan migrate:rollback --step=5
```

ROLLING BACK MIGRATIONS

- Câu lệnh migrate:reset sẽ thực hiện rollback lại toàn bộ migration của chương trình

```
php artisan migrate:reset
```

- Lệnh migrate:refresh đầu tiền sẽ rollback lại toàn bộ migration của chương trình, và thực hiện câu lệnh migrate. Câu lệnh sẽ thực hiện tái cấu trúc toàn bộ database:

```
php artisan migrate:refresh

php artisan migrate:refresh --seed
```

DEMO

Fpt University

FPT POLYTECHNIC

LARAVEL FRAMEWORK

BÀI 4 (PHẦN 2)

- Tạo table
  - Để tạo một bảng mới, sử dụng hàm create của Schema facacde. Hàm create nhận hai tham số. Tham số đầu là tên của bảng, còn tham số thứ hai là một Closure mà sẽ nhận vào một Blueprint object để khai báo cấu trúc của bảng mới
  - Schema::create('table_Name', function (Blueprint $table) { $table->increments('column_Name'); });

Tạo table

```php
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateDrinksTable extends Migration {

    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up() {
        Schema::create('drinks', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name',75)->unique();
            $table->text('comments')->nullable();
            $table->integer('rating');
            $table->date('brew_date');
            $table->timestamps();        });
    }
```

- Tạo table
  - Schema::create('drinks', function (Blueprint $table) {..}: sử dụng hàm tạo trong Schema class. function (Blueprint $table) sử dụng tham số $table định nghĩa cấu trúc database
  - $table->increments('id'): tham số tự tăng giá trị
  - $table->string('name',75)->unique(): kiểu string sử dụng cần lưu trữ chuỗi, số 75 là độ dài chuỗi và unique để xác định giá trị duy nhất
  - $table->text('comments')->nullable(): xác định text field và cho phép column nhận giá trị null

- Tạo table
  - $table->integer('rating'): xác định field kiểu int
  - $table->date('brew_date'): định nghĩa date field
  - $table->timestamps() : tạo 2 field time_stamp là created_at and updated_at
  - Chạy lệnh cmd: php artisan migrate để thực thi các câu lệnh migration
  - Kiểm tra kết quả trong PhpMyAdmin

- Kiểm tra xem bảng hay cột có tồn tại hay chưa
  - Sử dụng hasTable và hasColumn

```
if (Schema::hasTable('users')) {
    //
}

if (Schema::hasColumn('users', 'email')) {
    //
}
```

- Thực hiện thao tác schema trên một kết nối database không phải mặc định, sử dụng hàm connection:

```php
Schema::connection('foo')->create('users', function (Blueprint $table) {
    $table->increments('id');
});
```

Command	Description
`$table->engine = 'InnoDB';`	Specify the table storage engine (MySQL).
`$table->charset = 'utf8';`	Specify a default character set for the table (MySQL).
`$table->collation = 'utf8_unicode_ci';`	Specify a default collation for the table (MySQL).
`$table->temporary();`	Create a temporary table (except SQL Server).

- Renaming / Dropping Tables

```php
Schema::rename($from, $to);
```

```php
Schema::drop('users');

Schema::dropIfExists('users');
```

- Renaming Tables With Foreign Keys:Trước khi thay đổi tên bảng, nên kiểm tra xem có foreign key constraints nào trên bảng có tên khác trong migration file hay không thay vì để Laravel tự gán tên. Nếu không, tên của foreign key constraint sẽ trỏ tới tên cũ của table.

COLUMNS

- Tạo column

```
Schema::table('users', function (Blueprint $table) {
    $table->string('email');
});
```

- Column Types: bigIncrements, date, binary, decimal, enum, double,...
- Column Modifiers

```
Schema::table('users', function (Blueprint $table) {
    $table->string('email')->nullable();
});
```

Updating Column Attributes

```php
Schema::table('users', function (Blueprint $table) {
    $table->string('name', 50)->change();
});
```

```php
Schema::table('users', function (Blueprint $table) {
    $table->string('name', 50)->nullable()->change();
});
```

Renaming Columns

```php
Schema::table('users', function (Blueprint $table) {
    $table->renameColumn('from', 'to');
});
```

Dropping Columns

```php
Schema::table('users', function (Blueprint $table) {
    $table->dropColumn('votes');
});
```

Drop multiple columns

```php
Schema::table('users', function (Blueprint $table) {
    $table->dropColumn(['votes', 'avatar', 'location']);
});
```

FOREIGN KEY CONSTRAINTS

- Laravel cũng hỗ trợ cung cấp việc tạo foreign key constraint một cách dễ dàng. Ví dụ, cùng tạo một column user_id trên table posts tham chiếu tới column id trên bảng users:

```php
Schema::table('posts', function (Blueprint $table) {
    $table->integer('user_id')->unsigned();

    $table->foreign('user_id')->references('id')->on('users');
});
```

FOREIGN KEY CONSTRAINTS

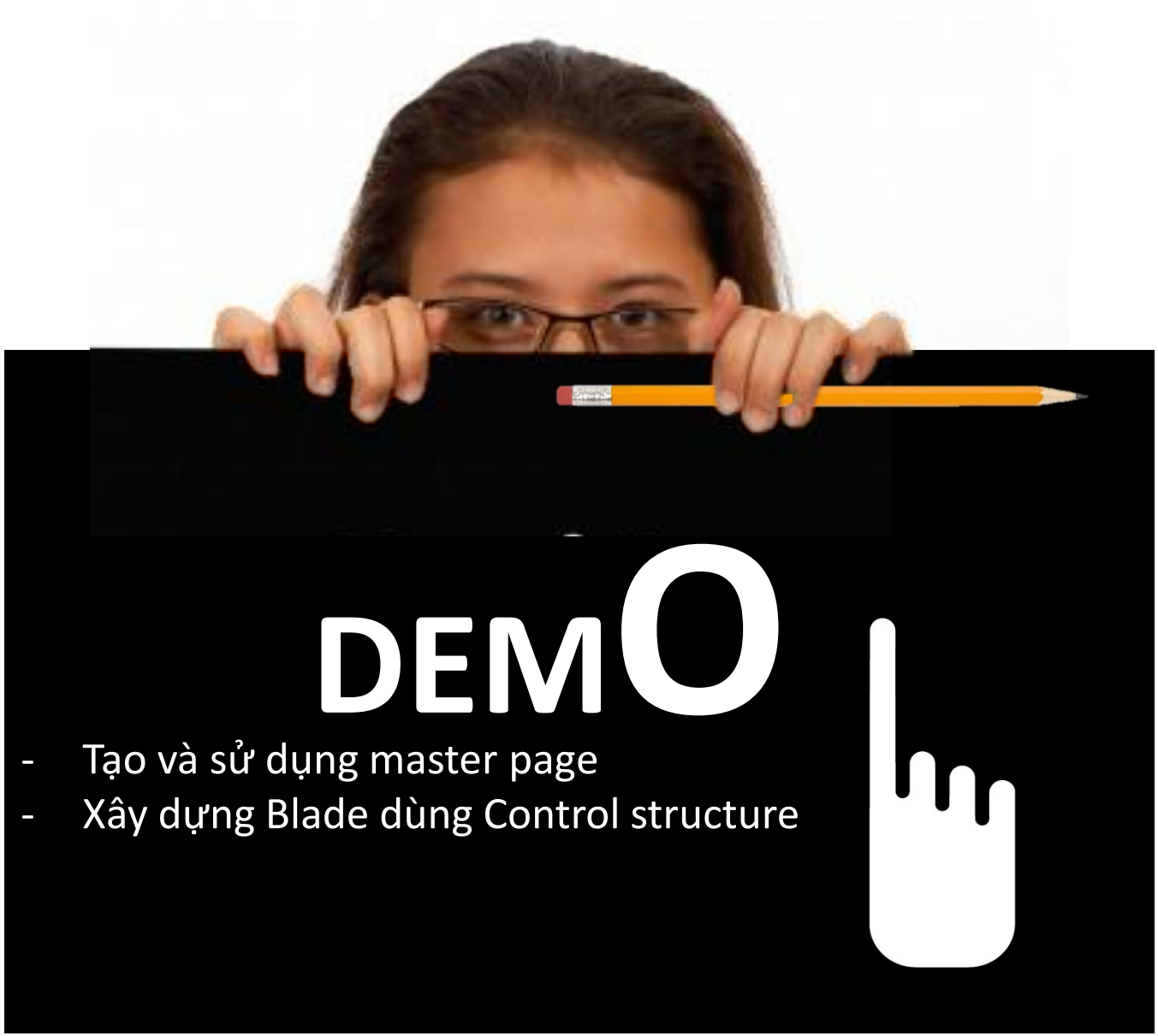
- Chỉ định thao tác cho thuộc tính "on delete" và "on update" của constraint

```
$table->foreign('user_id')
      ->references('id')->on('users')
      ->onDelete('cascade');
```

- Drop a foreign key

```
$table->dropForeign('posts_user_id_foreign');
```

```
$table->dropForeign(['user_id']);
```


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

TỔNG KẾT NỘI DUNG BÀI HỌC

- Phần I: Giới thiệu về Migration
  - Làm quen Migration
  - Các yêu cầu cần có khi chạy Migration
  - Cấu trúc và thực thi Migration
- Phần II: Tạo và làm việc với Migration
  - Tables
  - Columns
  - Foreign Key Constraints


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
Cảm ơn