**GIÁO TRÌNH LARAVEL - BUỔI 8**

**SCHEMA BULDER**

1. **Schema builder là gì ?**

*Schema là một Class trong Laravel, nó giúp Laravel có thể thao tác với các bảng dữ liệu như thêm, sửa, xóa một cách dễ dàng và khoa học.*

*Schema builder là cách nói về Laravel sử dụng các phương thức của Class Schema để thêm sửa, xóa các bảng và cột dữ liệu.*

1. **Tạo CSDL mẫu và kết nối với Laravel**

* **B1**. Vào PHPMyAdmin để tạo CSDL
* **B2**. Cấu hình kết nối CSDL chúng ta trong *.evn* và *app/config/database.php*

1. **Xây dựng bảng với Schema builder**

a. Phương thức tạo bảng dữ liệu create()

Cú pháp:

|  |
| --- |
| Schema::create(‘table\_name’, function($table){  // cú pháp tạo cột trong bảng  }); |

Ví dụ:

|  |
| --- |
| Schema::create('users', function($table){  $table->increments('user\_id'); //Tự tăng, khóa chính,int,unsigned  $table->string('email'); //Kiểu chuỗi varchar  $table->integer('phone'); //Kiểu int  $table->timestamps(); //Tự cập nhật thời gian  }); |

* Một số cú pháp tạo trường hay dùng :

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Miêu tả** |
| **$table->bigIncrements('id');** | Thuộc kiểu “BIGINT” và là khoá chính |
| **$table->bigInteger('votes');** | Thuộc kiểu “BIGINT” |
| **$table->binary('data');** | Thuộc kiểu BLOB |
| **$table->boolean('confirmed');** | Thuộc kiểu BOOLESE |
| **$table->char('name', 4);** | Thuôc kiểu CHAR(4) |
| **$table->date('created\_at');** | Thuôc kiểu Date |
| **$table->dateTime('created\_at');** | Thuôc kiểu DateTime |
| **$table->decimal('amount', 5, 2);** | Thuôc kiểu DECIMAL |
| **$table->double('column', 15, 8);** | Thuôc  double kiểu tương đương với độ chính xác, tổng cộng 15 chữ số và 8 chữ số sau dấu thập phân |
| **$table->float('amount');** | Thuôc kiểu FLOAT |
| **$table->increments('id');** | Tự tăng, khóa chính,int,unsigned |
| **$table->integer('votes');** | Thuôc kiểu INT |
| **$table->json('options');** | Thuôc kiểu JSON |
| **$table->jsonb('options');** | Thuôc kiểu JSONB |
| **$table->longText('description');** | Thuôc kiểu LONGTEXT |
| **$table->mediumInteger('numbers');** | Thuôc kiểu MEDIUMINT |
| **$table->mediumText('description');** | Thuôc kiểu MEDIUMTEXT |
| **$table->smallInteger('votes');** | Thuôc kiểu SMALLINT |
| **$table->tinyInteger('numbers');** | Thuôc kiểu TINYINT |
| **$table->string('email');** | Thuôc kiểu VARCHAR (255) |
| **$table->string('name', 100);** | Thuôc kiểu VARCHAR (100) |
| **$table->text('description');** | Thuôc kiểu Text |
| **$table->time('sunrise');** | Thuôc kiểu Time |
| **$table->timestamp('added\_on');** | Thuôc kiểu TIMESTAMP |
| **$table->timestamps();** | Thêm các 2 cột created\_at và update\_at |
| **$table->rememberToken();** | Thêm cột remember\_token dưới dạng VARCHAR (100) NULL |

b. Điều kiện cho trường trong bảng

|  |  |
| --- | --- |
| Câu lệnh | Mô tả |
| ->nullable(); | Cho phép giá trị null |
| ->default($value) | Gán giá trị mặc định cho cột |
| ->unsigned() | Đặt unsigned cho interger |

Ví dụ:

Schema::create('users', function($table){

$table->increments('user\_id'); //Tự tăng, khóa chính,int,unsigned

$table->string('TenSanPham')->nullable(); //Kiểu chuỗi varchar, có thể null

});

c. Khóa ngoại (Foreign key)

*Foreign key là mối quan hệ giữa hai bảng và mối quan hệ này ta hay gọi là cha - con, nghĩa là nếu bảng A có một thuộc tính liên kết tới bảng B thì lúc này bảng B đóng vai trò là cha và bảng A đóng vai trò là con.*

Cú pháp:

$table->interger('user\_id')->unsigned();

$table->foreign('user\_id')->references('id')->on('users');

// liên kết có ràng buộc giữa 2 bảng (khi xóa bản ghi ở bảng có khóa chính thì bản ghi ở bảng chứa khóa ngoại liên kết với khóa chính đó bị xóa theo) :

$table->foreign('user\_id')->references('id')->on('users')->onDelete('cascade');

d. Phương thức đổi tên bảng dữ liệu rename()

Cú pháp:

|  |
| --- |
| Schema::rename(oldTableName, newTableName); |

Ví dụ:

|  |
| --- |
| Schema::rename(‘users’, ‘thanhvien’); |

e. Phương thức xóa bảng dữ liệu drop()

Cú pháp:

|  |
| --- |
| Schema::drop(tableName);  Schema::dropIfExists(tableName); |

Ví dụ:

|  |
| --- |
| Schema::drop('users');  Schema::dropIfExists('users'); |

f. Phương thức thêm một cột mới vào bảng table()

Cú pháp:

|  |
| --- |
| Schema::table(tableName, function($table){  $table->string(columnName);  }); |

Ví dụ:

|  |
| --- |
| Schema::table('users', function($table){  $table->string(user\_name);  }); |

g. Thay đổi thuộc tính cột

Để thay đổi thuộc tính của cột ta cần phâỉ cập nhật thư viện doctrine/dbal bằng cách nhập câu lệnh trong cửa sổ terminal :

**composer require doctrine/dbal**

Các phương thức :

* Phương thức đổi tên cột renameColumn()

Cú pháp:

|  |
| --- |
| Schema::table(tableName, function($table){  $table->renameColumn(oldColumnName, newColumnName);  }); |

Ví dụ:

|  |
| --- |
| Schema::table(‘users’, function($table){  $table->renameColumn(‘user\_id’, ‘userid’);  }); |

* Phương thức thêm một hoặc nhiều cột :

Schema::table('users', function ($table) {

$table->string('ok', 255)->nullable();

});

* Phương thức xóa một hoặc nhiều cột từ bảng dropColumn()

Cú pháp:

|  |
| --- |
| Schema::table(tableName, function($table){  $table->dropColumn(columnName);  });  Schema::table(tableName, function($table){  $table->dropColumn(columnName-1, columnName-2,… columnName-N);  }); |

Ví dụ:

|  |
| --- |
| Schema::table('users', function($table){  $table->dropColumn('user\_name');  });  Schema::table('users', function($table){  $table->dropColumn('user\_name', 'user\_pass', 'user\_mail');  }); |

f. Phương thức kiểm tra sự tồn tại của bảng hasTable()

Cú pháp:

|  |
| --- |
| if(Schema::hasTable(tableName)){  //  } |

Ví dụ:

|  |
| --- |
| if(Schema::hasTable('users')){  //  } |

g. Phương thức kiểm tra sự tồn tại của cột hasColumn()

Cú pháp:

|  |
| --- |
| if(Schema::hasColumn(tableName, columnName)){  //  } |

Ví dụ:

|  |
| --- |
| if(Schema::hasColumn('users', 'user\_name')){  //  } |