1. Migrations

Installation d’un nouveau projet Laravel

Creation d’un virtual host

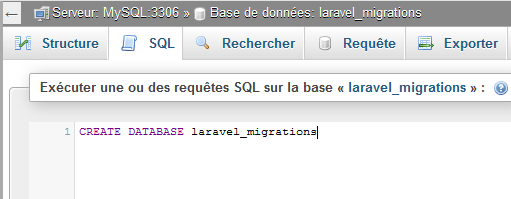
Modifier le fichier config>database.php



Modifier le fichier **.ENV** comme ci-dessous

DB\_DATABASE=laravel\_migration  
DB\_USERNAME=test  
DB\_PASSWORD=blabla

Créer une base de donnée dans MySQL



Dans le terminal se placer dans notre dossier de travail et taper :

*php artisan migrate*

La BDD se remplit avec les champs par defaut de Laravel



Dans le dossier migrations

Dans la fonction « up »

public function up()  
{  
 Schema::create('posts', function (Blueprint *$table*) {  
 *$table*->increments('id'); //tu vas me creer un champs de type id  
 *$table*->string('title');  
 *$table*->text('content');  
  
 *$table*->timestamps();  
 });  
}

Dans la fonction « down » corriger comme ci-dessous

public function down()  
{  
 Schema::drop('posts');  
}

Dans le terminal taper :

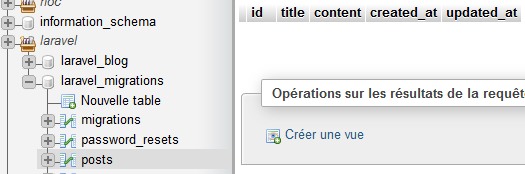
*php artisan migrate*

*$ php artisan migrate*

*Migrating: 2018\_06\_21\_143630\_create\_posts\_table*

*Migrated: 2018\_06\_21\_143630\_create\_posts\_table*

On vient de créer une Table dans notre BDD



**CREATION DES MIGRATION POUR LE BLOG**

Les Tables que l’on veut créer :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ROLE** |  | **PHOTO** |  | **POSTS** |  | **CATEGORIES** |
| id |  | id |  | id |  | id |
| name |  | filename |  | title |  | name |
| time |  | time |  | content |  | time |
|  |  |  |  | user\_id |  |  |
|  |  |  |  | photo\_id |  |  |
|  |  |  |  | category\_id |  |  |
|  |  |  |  |  |  |  |
|  |  | **USERS** |  | **COMMENTS** |  |  |
|  |  | id |  | id |  |  |
|  |  | name |  | post\_id |  |  |
|  |  | email |  | is\_active |  |  |
|  |  | password |  | content |  |  |
|  |  | remenber\_token |  | author |  |  |
|  |  | time |  | email |  |  |
|  |  | role\_id |  | time |  |  |
|  |  | is\_active |  |  |  |  |

// NORMALEMENT TOUTES MIGRATIONS ONT DEJA ETES CREER LORS DE LA CREATION DES MODELS (*php artisan make:model MonModel* **–m**)

Si ce n’est pas le cas faire :

Dans un terminal on créer nos migrations :

php artisan make:migration create\_roles\_table --create=roles



php artisan make:migration create\_photos\_table --create=photos



php artisan make:migration create\_posts\_table --create=posts



php artisan make:migration create\_comments\_table --create=comments



php artisan make:migration create\_categories\_table --create=categories

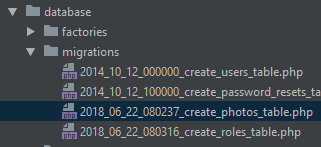


On rajoute une migration role\_user pour faire notre liaison pivot ManyToMany (voir les Models)

php artisan make:migration create\_role\_user\_table --create=role\_user



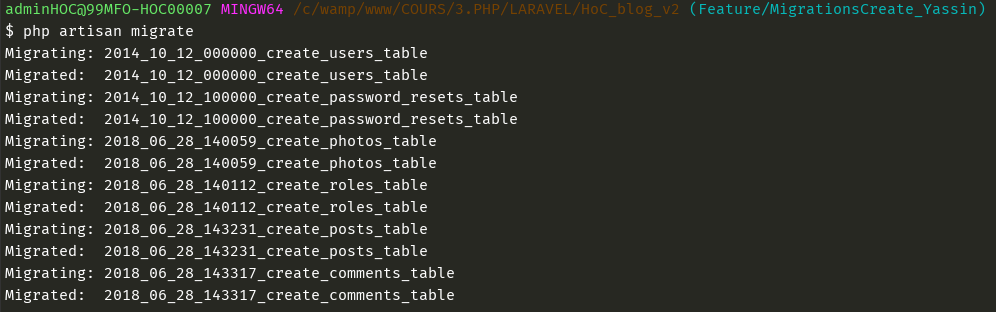
Resultat :



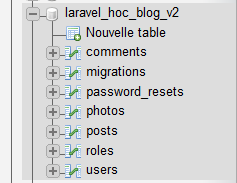
(Pour revenir en arrière sur une migration, dans un terminal taper

« composer dump-autoload » )

On fait un php artisan migrate et on vérifie le contenu de notre BDD



Dans la BDD on a généré nos tables



**Ensuite on remplit nos colonnes pour chaque table :**

**Roles**

|  |
| --- |
| **ROLE** |
| id |
| name |
| time |

public function up()  
{  
 Schema::create('roles', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('name');  
 *$table*->timestamps();  
 });  
}

**Roles\_user (fonction pivot : Roles->Users)**

public function up()  
{  
 Schema::create('role\_user', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table* ->integer('role\_id')->index()->unsigned();  
 *$table*->integer('user\_id')->index()->unsigned();  
 *$table*->timestamps();  
 });  
}

**Photos**

|  |
| --- |
| **PHOTO** |
| id |
| filename |
| time |

public function up()  
{  
 Schema::create('photos', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('filename');  
 *$table*->timestamps();  
 });  
}

**Posts**

|  |
| --- |
| **POSTS** |
| id |
| title |
| content |
| user\_id |
| photo\_id |
| category\_id |

public function up()  
{  
 Schema::create('posts', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('title');  
 *$table*->text('content');  
 *$table*->integer('user\_id');  
 *$table*->integer('photo\_id');  
 *$table*->integer('category\_id');  
 *$table*->timestamps();  
 });  
}

**Categories**

|  |
| --- |
| **CATEGORIES** |
| id |
| name |
| time |

public function up()  
{  
 Schema::create('categories', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('name');  
 *$table*->timestamps();  
 });  
}

**Users**

|  |
| --- |
| **USERS** |
| id |
| name |
| email |
| password |
| remenber\_token |
| time |
| role\_id |
| is\_active |

public function up()  
{  
 Schema::create('users', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('name');  
 *$table*->string('email');  
 *$table*->string('password');  
 *$table*->rememberToken();  
 *$table*->boolean('is\_active');  
 *$table*->timestamps();  
 });  
}

On va faire nos jointures sur la jonction « role\_id

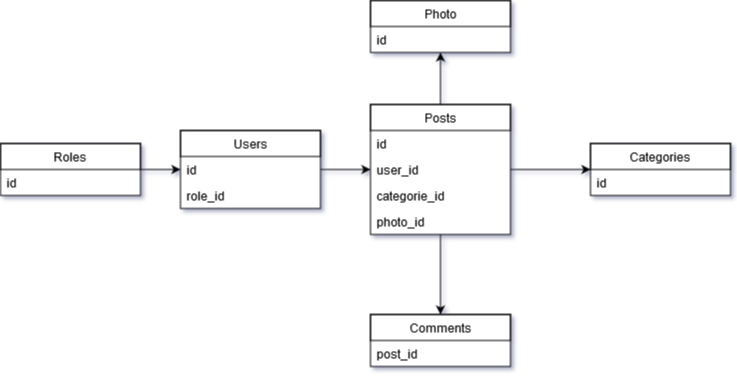
**Comments**

|  |
| --- |
| **COMMENTS** |
| id |
| post\_id |
| is\_active |
| content |
| author |
| email |
| time |

public function up()  
{  
 Schema::create('comments', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->integer('post\_id');  
 *$table*->text('author');   
 *$table*->string('email');  
 *$table*->longText('content');  
 *$table*->boolean('is\_active');  
 *$table*->timestamps();  
 });

**Contraintes :**

Et on rajoute nos contraintes à nos tables :



**Roles**

|  |
| --- |
| **ROLES** |
| id |
| name |
| time |

public function up()  
{  
 Schema::create('roles', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('name');  
 *$table*->timestamps();  
 });  
}

**Photos**

|  |
| --- |
| **PHOTOS** |
| id |
| filename |
| time |

public function up()  
{  
 Schema::create('photos', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('file');  
 *$table*->timestamps();  
 });  
}

**Posts**

|  |
| --- |
| **POSTS** |
| id |
| title |
| content |
| user\_id |
| photo\_id |
| category\_id |
| Is\_active |
| time |

**POSTS** :

user\_id = doit être index et unsigned

photo\_id = doit être index et unsigned

category\_id = doit être index et unsigned

public function up()  
{  
 Schema::create('posts', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('title');  
 *$table*->text('content');  
 *$table*->integer('user\_id')->index()->unsigned();  
 *$table*->integer('photo\_id')->index()->unsigned();  
 *$table*->integer('category\_id')->index()->unsigned();  
 *$table*->boolean('is\_active')->default(0);  
 *$table*->timestamps();  
 *$table*->foreign('user\_id')->references('id')->on('users')->onDelete('cascade');  
 });  
}

**Users**

|  |
| --- |
| **USERS** |
| id |
| name |
| email |
| password |
| remenber\_token |
| time |
| role\_id |
| is\_active |

**USERS** :

role\_id = doit être un index, unsigned, nullable

is\_active = doit être par defaut à 0

email = unique

public function up()  
{  
 Schema::create('users', function (Blueprint *$table*) {  
 *$table*->increments('id';  
 *$table*->string('name');  
 *$table*->string('email')->unique();  
 *$table*->string('password');  
 *$table*->rememberToken();  
 *$table*->boolean('is\_active')->default(0);  
 *$table*->timestamps();  
 });  
}

On va faire nos jointures sur la jonction « role\_id »

**Comments**

|  |
| --- |
| **COMMENTS** |
| id |
| post\_id |
| is\_active |
| content |
| author |
| email |
| time |

**COMMENTS** :

post\_id = doit être index et unsigned

is\_active = par defaut à 0

public function up()  
{  
 Schema::create('comments', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->integer('post\_id')->index()->unsigned();  
 *$table*->text('author');   
 *$table*->string('email');  
 *$table*->longText('content');  
 *$table*->boolean('is\_active')->default(0);  
 *$table*->timestamps();  
 *$table*->foreign('post\_id')->references('id')->on('posts')->onDelete('cascade');  
 });

**Password**

**On supprime ->index() pour le ‘token’**

public function up()  
{  
 Schema::create('password\_resets', function (Blueprint *$table*) {  
 *$table*->string('email')->index();  
 *$table*->string('token');  
 *$table*->timestamp('created\_at')->nullable();  
 });  
}

**Categories**

|  |
| --- |
| **CATEGORIES** |
| id |
| name |
| time |

public function up()  
{  
 Schema::create('categories', function (Blueprint *$table*) {  
 *$table*->increments('id');  
 *$table*->string('name')->unique();  
 *$table*->timestamps();  
 });  
}

**Clés étrangères :**

On ajoute les clés étrangères

Créer une clé foreign pour Posts, Comments

*$table*->foreign('post\_id')->references('id')->on('posts')->onDelete('cascade');

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

Vue finale dans les fichiers :

