

## Travaux Pratiques n° 8

### Laravel : Utilisation des bases de données

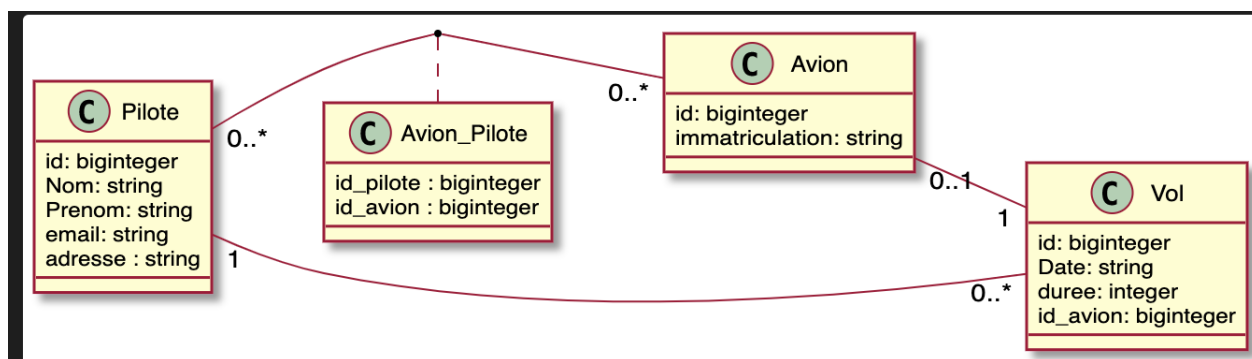
## 1 Les ressources

Dans ce Tp nous allons voir comment mettre en œuvre les différents types de liaisons que l'on peut rencontrer entre les tables d'une base de données. Pour cela nous allons construire un site de gestion d'un aéroclub, en partant d'une distribution vierge de Laravel

### 1.1 Le cahier des charges

Le site doit permettre de gérer le profil des pilotes (nom, prénom, email, adresse, etc). Chaque pilote peut effectuer des vols. Un vol est défini par sa date, sa durée, le pilote qui l'a réalisé, l'avion utilisé, ... Un avion est quant à lui défini par son immatriculation et son type (le modèle). On limite ici volontairement le nombre de champs des différentes tables. Celle-ci pourront ensuite être complétées si nécessaire.

Pour pouvoir prendre un avion, le pilote doit être qualifié pour cette appareil. Un pilote peut être qualifié sur un ou plusieurs appareils. Réciproquement, un avion peut être piloté par un ou plusieurs pilotes. La figure suivante propose un MCD simplifié de la base.



### 1.2 Table pilote.

Mettre en place la table pilote, et construisez le contrôleur CRUD permettant de réaliser les différentes actions élémentaires de gestion de cette table. Pour le moment on ne s'occupe pas des autres tables. Vous pourrez pour votre migration vous inspirer du code suivant :

```

public function up()
{
    Schema::create('pilotes', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('nom');
        $table->string('prenom');
        $table->string('email')->unique();
        $table->string('adresse');
        $table->timestamps();
    });
}

```

La figure suivante donne un aperçu de la structure de la page permettant d'effectuer les différentes opérations de base de gestion d'un pilote.

Liste Pilotes				Créer un pilote
#	Nom			
1	Dayton	Voir	Modifier	Supprimer
2	Edwardo	Voir	Modifier	Supprimer
7	Gennaro	Voir	Modifier	Supprimer
9	Magnus	Voir	Modifier	Supprimer
4	Myrtis	Voir	Modifier	Supprimer

< 1 2 >

Enfin pour vous faciliter la tâche durant la phase de test vous pouvez créer une factory (<https://laravel.com/docs/6.x/database-testing#writing-factories>) en vous inspirant du code suivant :

```

$factory->define(Pilote::class, function (Faker $faker) {
    return [
        'nom' => $faker->firstName,
        'prenom' => $faker->lastName,
        'email' => $faker->email,
        'adresse' => $faker->address
    ];
});

```

## 1.3 Gestion des vols : relation one to many

Nous voulons ajouter à la vue de chaque fiche pilote, la liste de ses vols comme sur la figure suivante par exemple.

Nom : Myrtis			
Prénom : Kohler			
Email : smith.delfina@yahoo.com			
Adresse : 7366 West Oval West Domenico, AL 70124-3467			
Date	Heure de départ	Durée	Fonction
2011-04-26	06:39:00	05:49:00	I
2001-10-16	09:07:00	16:24:00	CB
2018-05-18	20:02:00	13:59:00	EP
1987-10-25	13:35:00	06:09:00	I
1997-01-17	03:33:00	05:52:00	CB
<div>&lt; 1 2 &gt;</div>			

Pour cela vous allez devoir mettre en place une relation de type 1 :n. Pour vous aider vous pouvez vous reporter aux liens suivants :

- <https://laravel.com/docs/6.x/eloquent-relationships#defining-relationships>
- <https://laravel.sillo.org/cours-laravel-6-les-donnees-la-relation-1n/>

### Travail à effectuer

- Mettre en place la table pilote ;
- Créer une factory et un seeder pour alimenter cette nouvelle table ;
- Mettre en place les relations Pilote vers Vol (`hasMany()`) et Vol vers Pilotes (`belongsTo()`).
- Modifier votre template de visualisation du profile pilote

## 1.4 Gestion des avions : relation many to many

Nous allons maintenant compléter notre application en ajoutant la gestion des avions comme présentée dans le cahier des charges. Pour effectuer cette modification vous pouvez vous reporter aux liens suivants :

- <https://laravel.com/docs/6.x/eloquent-relationships#many-to-many>
- <https://laravel.sillo.org/cours-laravel-6-les-donnees-la-relation-nn/>