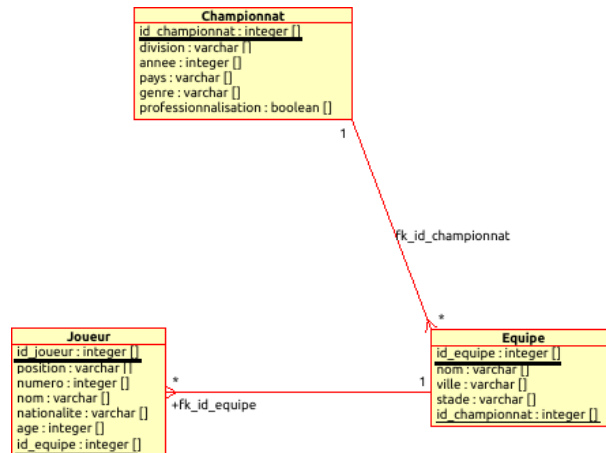


## Contrôle 5

### TP Football

L'application «TPFootball» est basée sur le diagramme de relation d'entité suivant :



En vous basant sur l'exemple pureMVC, je vous demande de :

- Créer les tables avec PHP
- Créer les Modèles correspondant avec leur Getter et Setter méthodes
- Créer les Contrôleurs de ces modèles avec un CRUD
- Créer des contrôleurs gérant les balises HTML, les Formulaires et les Styles.
- Créer les Vues pour chaque Modèle avec un CRUD dans un dossier « /Views/ » contenant les fichiers `championnat.lame.php`, `joueur.lame.php` et `equipe.lame.php`
- Implémenter les CRUD dans les Vues grâce à des Formulaires
- Créer un Menu de navigation en utilisant `HtmlController` dans « /Routes/Web.php » que l'on peut appeler avec «`$web = new Web()`» dans les pages de vue (« Routes » est un nouveau dossier dans la racine du serveur)
- Ajouter un champ « photo » aux Joueurs et un « blason » aux Equipes et Championnats qui permet d'associer et d'afficher une photo (ou blason) à chaque instance

The screenshot shows a web form titled "Football Registration Form" with a background image of a soccer field. The form contains the following fields:

- Player Name:** Two input fields labeled "First" and "Last".
- Player Email:** One input field.
- Player Team Group:** A dropdown menu with the text "\*Please Select\*" and a downward arrow.
- Offense or Defense:** Three radio buttons labeled "Offense", "Defense", and "Don't Know".
- Height:** A label "Height in inches" followed by an input field.
- Weight:** A label "Weight in lbs" followed by an input field.
- 40-Yard Dash:** An input field.
- Comments:** A text area at the bottom.

- Créer une page spécifique (par exemple `equipe_joueur.lame.php`) qui réalise une jointure/vue de deux tables (Joueur + Equipe) affichant chaque joueur avec son équipe associée

- Rendre toutes les classes PHP accessibles grâce à Autoload.php et « namespace »
- Au sein de MAMP ou toute autre serveur et BDD, de pouvoir tester la bonne fonctionnalité de tout le CRUD pour chaque modèle.

Attention à rajouter ces lignes de code afin de bénéficier de l'implémentation de clefs étrangères dans SQLite :

Dans les créations de table :

```
FOREIGN KEY(id_<entite>) REFERENCES <entites>(id) ON DELETE CASCADE
```

Dans Database.php

```
$this->query("PRAGMA foreign_keys = ON");
```

Je vous laisse libre du détail de chaque tâche, néanmoins, je vous conseille de toujours garder en tête l'utilisation de la POO en parallèle avec la structure MVC, autant que possible.

Bon codage estival !