

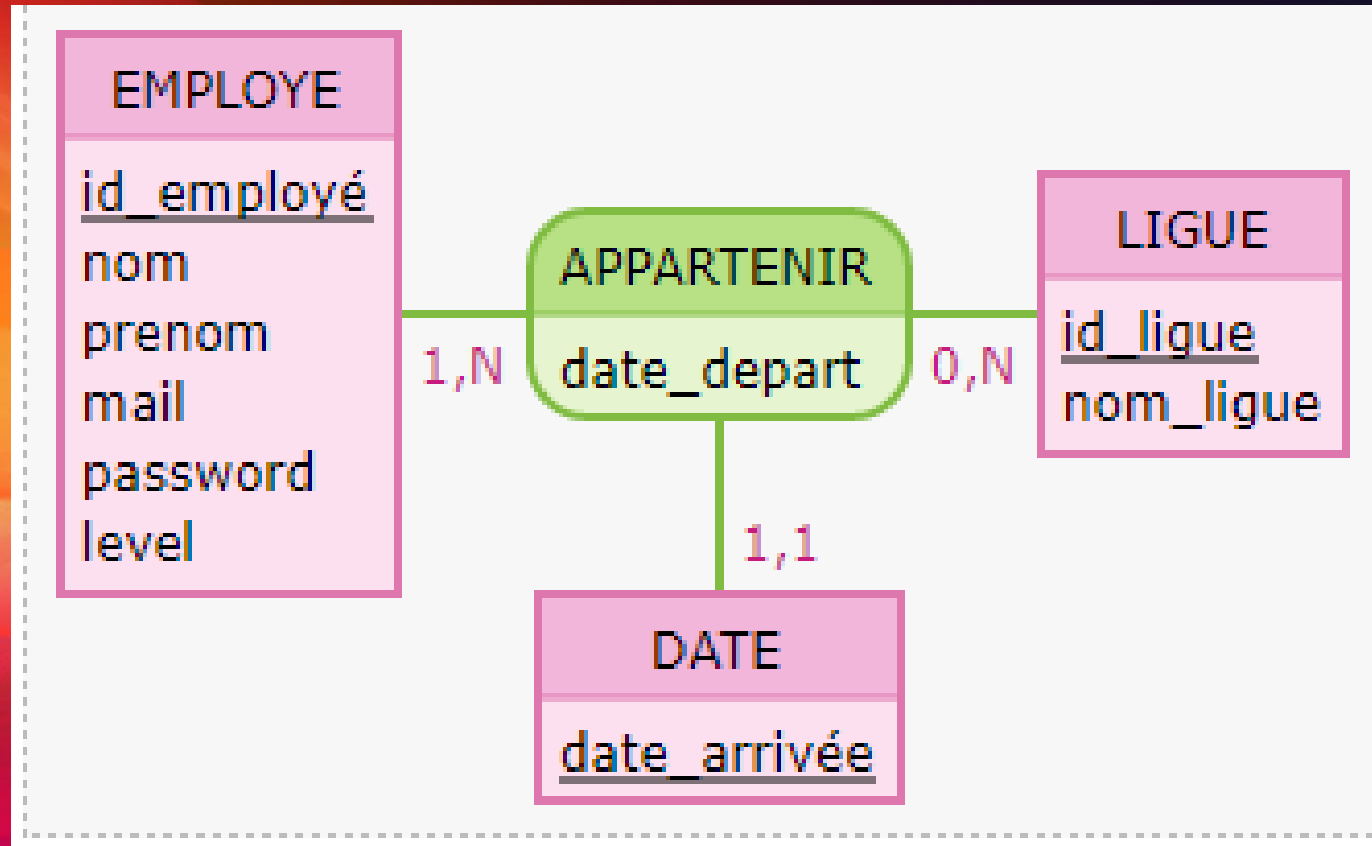
PPE - GESTION DU PERSONNEL DES LIGUES

La gestion des inscriptions aux compétitions est encore aujourd'hui assurée par les ligues avec des tableurs, et le développement d'un logiciel davantage adapté est en cours.

L'application en question permet de gérer un ensemble de compétitions, de personnes, et d'affecter des personnes à des compétitions. Il est possible que certaines compétitions soient réservées à des équipes et qu'il soit impossible à une personne seule de s'inscrire dans le cas, tous les membres de l'équipe doivent être enregistrés.

Cette application ne sera pas accessible au grand public, seuls des employés des ligues pourront y accéder pour saisir les inscrits. Les précédents informaticiens se sont penchés sur le problème et ont déjà commencé à développer la couche métier

PPE - GESTION DU PERSONNEL DES LIGUES



Un employé appartient à 1 seule ligue.

1 ligue peut avoir aucun 1 ou plusieurs employés. Et a un seul administrateur.

Le super-administrateur a accès en écriture à tous les employés des ligues.

PPE - GESTION DU PERSONNEL DES LIGUES

PARTAGE DE TACHES

The screenshot displays a Trello board interface for a project named 'personnel'. The board is organized into four columns: 'A faire' (To do), 'En cours' (In progress), 'Terminé' (Completed), and 'A compl' (Completed). The 'En cours' column is currently active, showing a task card titled 'faire la gestion des dates'. This card has a progress bar with a green segment and an orange segment, a red notification bell icon with the number '2', and a visibility icon. It also features 'S' and 'YN' status buttons. Below the card is a button to '+ Ajouter une autre carte'. The 'Terminé' column shows two task cards: 'testUnitaires' and 'faire le MCD', both with progress bars and notification icons. The 'faire le MCD' card also has a visibility icon. The 'faire un arbre heuristique' card is partially visible at the bottom of the 'Terminé' column. The board's header includes a search bar, the Trello logo, and a list of board members: 'Butler', 'S', 'YN', and 'Inviter'. The background of the board is a landscape image of a mountain range.

LES TESTS UNITAIRES

Un test unitaire est une procédure visant à vérifier le bon fonctionnement d'un programme.

```
package testsUnitaires;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Test;

import personnel.*;

class testLigue
{
    @Test
    void testCreateLigue()
    {
        Ligue ligue = new Ligue("Fléchettes");
        assertEquals("Fléchettes", ligue.getNom());
    }

    @Test
    void testAddEmploye()
    {
        Ligue ligue = new Ligue("Fléchettess");
        Employe employe = ligue.addEmploye("Bouchard", "Gérard", "g.bouchard@gmail.com", "azerty");
        assertEquals(employe, ligue.getEmployes().first());
    }
}
```

GESTION DES DATES

GESTION DE LA DATE DE DÉPART ET DE CELLE D'ARRIVÉE DE CHAQUE EMPLOYÉ

@Test

```
void testGetArrive() {  
    Ligue ligue = new Ligue("Fléchettes");  
    LocalDate dateArrive = LocalDate.of(2020, 01, 01);  
    Employe employe = ligue.addEmploye("selima", "bk", "sbk@gmail.com", "pass", dateArrive);  
    LocalDate date = LocalDate.of(2020, 01, 21);  
    employe.setDateDepart(date);  
    assertEquals(employe.getDateArrive(), dateArrive);  
}
```

@Test

```
void testGetDepart() {  
    Ligue ligue = new Ligue("Fléchettes");  
    Employe employe = ligue.addEmploye("selima", "bk", "sbk@gmail.com", "pass", null);  
    LocalDate date = LocalDate.of(2020, 01, 21);  
    employe.setDateDepart(date);  
    assertEquals(employe.getDateDepart(), date);  
}
```

}