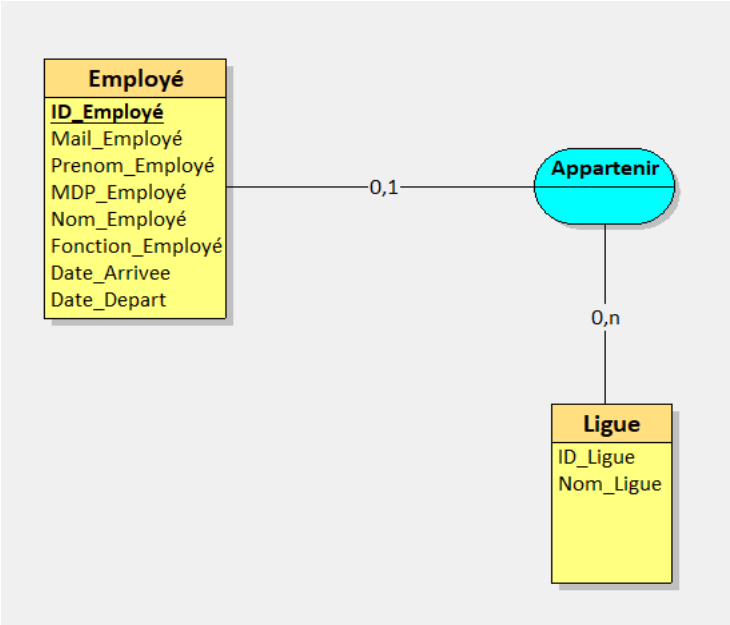
Compte rendu Projet Personnel itération 2

Cette application permet de gérer des ligues et des employés via une interface en ligne de commande. Elle est conçue pour permettre à un administrateur de créer et de gérer des ligues, d'ajouter, de modifier et de supprimer des employés dans ces ligues, ainsi que de gérer l'administrateur de chaque ligue.

Voici les fonctionnalités de chaque classes :

- Les principales entités du système sont les ligues et les employés**,** et l'application offre un ensemble de fonctionnalités pour interagir avec ces entités.
- La classe Ligue représente une ligue à laquelle les employés peuvent appartenir. Une ligue possède un nom et peut être administrée par un employé, qui a des droits spécifiques sur la gestion des employés de la ligue.
- La classe Employe représente un employé travaillant dans une ligue. Un employé a des informations personnelles telles que son nom, son prénom, son email, un mot de passe, ainsi que des dates d'arrivée et de départ dans la ligue.
- La classe EmployeConsole sert à gérer l'interface de ligne de commande pour interagir avec les employés. Elle permet à l'utilisateur de manipuler les informations des employés via un menu.
- La classe LigueConsole est utilisée pour gérer l'interface de ligne de commande pour interagir avec les ligues. Elle permet à l'utilisateur d'interagir avec les ligues en fournissant un menu interactif pour gérer leurs informations et leurs employés.

Sommaire :



Un employé peut appartenir à une seule ligue.

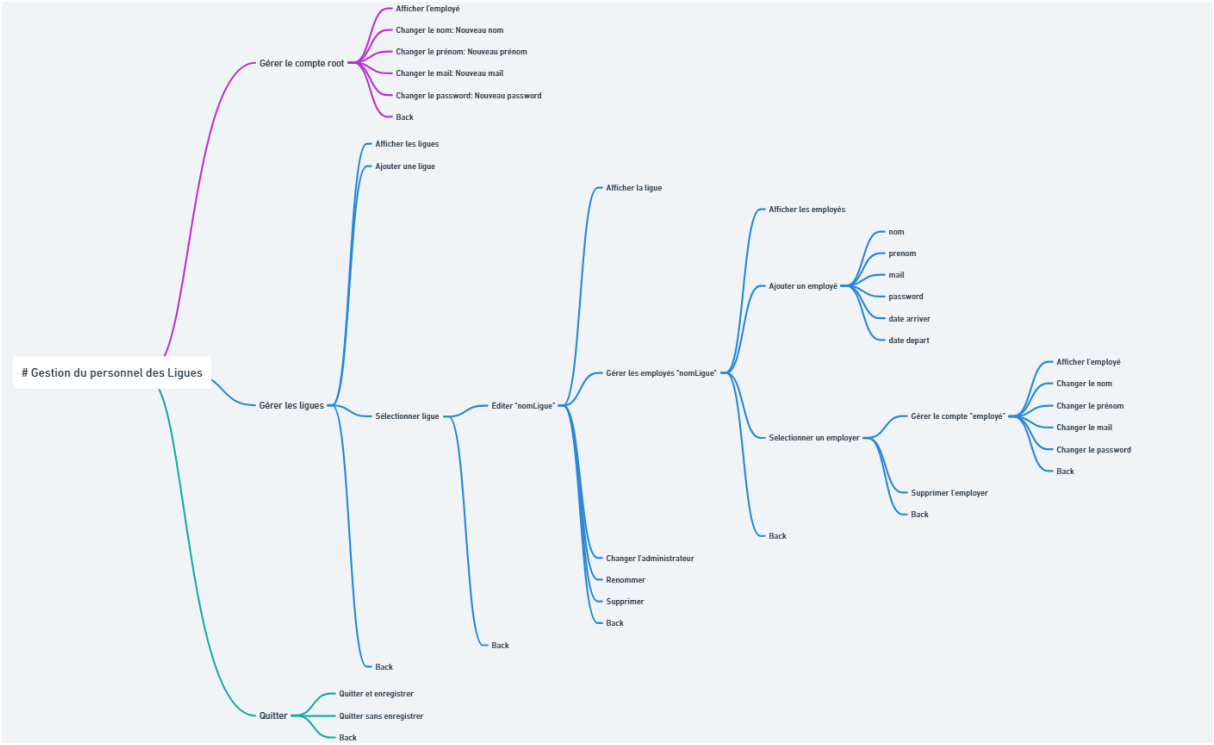
Une ligue peut être composé de plusieurs Employé.

Voici le Script : base de données

```sql
CREATE TABLE Ligue (
    ID_Ligue INT ,
    Nom_Ligue VARCHAR(50),
    PRIMARY KEY(ID_Ligue)
);


CREATE TABLE Employé (
    ID_Employé INT ,
    Nom_Employé VARCHAR(20),
    Prenom_Employé VARCHAR(20),
    Mail_Employé VARCHAR(50),
    MDP_Employé VARCHAR(50),
    Fonction_Employé VARCHAR(50),
    Date_Depart DATE,
    Date_Arrivee DATE,
    ID_Ligue INT,
    PRIMARY KEY(ID_Employé),
    FOREIGN KEY(ID_Ligue) REFERENCES Ligue(ID_Ligue)
);
```

Arbre heuristique V3 :



Code Java : Changement administrateur

```
private Menu editerLigue(Ligue ligue)
{
    Menu menu = new Menu("Editer " + ligue.getNom());
    menu.add(afficher(ligue));
    menu.add(gererEmployes(ligue));
    menu.add(changerAdministrateur(ligue))
    menu.add(changerNom(ligue));
    menu.add(supprimer(ligue));
    menu.addBack("q");
    return menu;
}
```

```
private List<Employe> changerAdministrateur(final Ligue ligue)
{
    return list<>("changer l'administrateur", "c"
    () -> new ArrayList<>(ligue.getEmployes())
    (index, element) -> {ligue.setAdministrateur(element);System.out.println("Le nouvel administrateur est : " + element)}
    );
}
```

Code Java :Employé

```
public class Employe implements Serializable, Comparable<Employe>
{
    private static final long serialVersionUID = 4795721718037994734L;
    private String nom, prenom, password, mail;
    private Ligue ligue;
    private GestionPersonnel gestionPersonnel;
    private LocalDate dateArrive;
    private LocalDate dateDepart;
```

```
Employe(GestionPersonnel gestionPersonnel, Ligue ligue, String nom, String prenom, String mail, String password, LocalDate dateArrive, LocalDate dateDepart)
        throws ExceptionD
{
    this.gestionPersonnel = gestionPersonnel;
    this.nom = nom;
    this.prenom = prenom;
    this.password = password;
    this.mail = mail;
    this.ligue = ligue;
```

```
public void setDateArrivee(LocalDate dateArrive)
        throws ExceptionD
{
    if (dateArrive == null  || dateDepart.isBefore(dateArrive)) {
        throw new ExceptionD();
    }
    else {

        this.dateArrive = dateArrive;}
}
```

```
    if (dateArrive == null || dateDepart == null || dateDepart.isBefore(dateArrive) ) {
        throw new ExceptionD();
    }

    this.dateArrive = dateArrive;
    this.dateDepart = dateDepart;
```

```
public LocalDate getDateDepart()
{
    return this.dateDepart;
}
```

```
public LocalDate getDateArrivee()
{
    return this.dateArrive;
}
```

Code Java : Ligue

```java
import java.time.LocalDate;
```

```java
public Employe addEmploye(String nom, String prenom, String mail, String password , LocalDate dateArrivee , LocalDate Datedepart)
throws ExceptionD
{

    Employe employe = new Employe(this.gestionPersonnel, this, nom, prenom, mail, password , dateArrivee , Datedepart);
    employes.add(employe);
    return employe;
}
```