

AP - Personnel

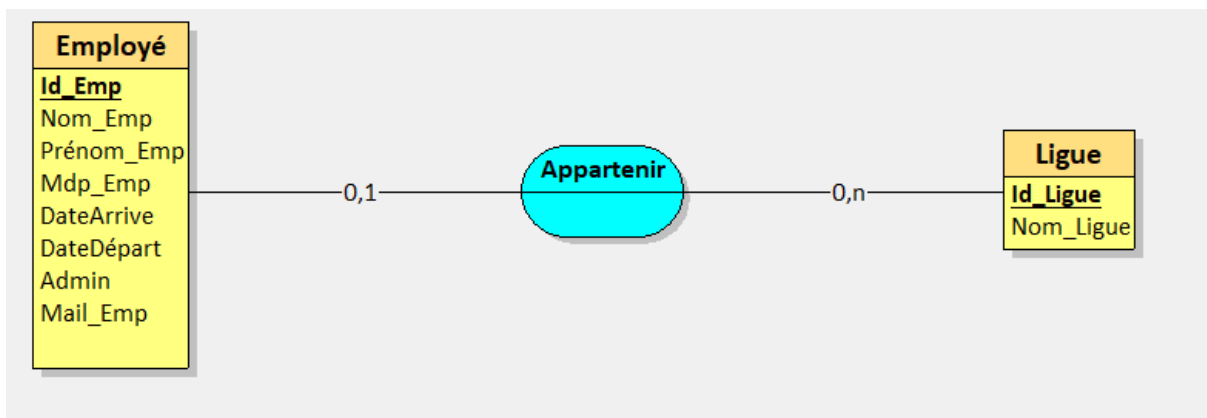
Avec notre groupe d'AP nous avons modifié une application Java réalisé par notre professeur en faisant un Fork via git (git clone)

```
KADEY@DESKTOP-J3AFIU5 MINGW64 ~/Desktop/Kadeyofficiel.github.io (main)
$ git clone + adresse du dépôt
```

Les modifications réalisés pour l'instant sont :

- + La gestion des ligues (insertion, modification, suppression)
- + La gestion des employés (insertion, modification, suppression)
- + La gestion de l'administrateur (en ligne de commande/interface du terminal)
 - + La gestion des dates, mot de passes, tests unitaires
- + La création la base de données correspondante et du lien entre les deux (NON FINALISÉ)

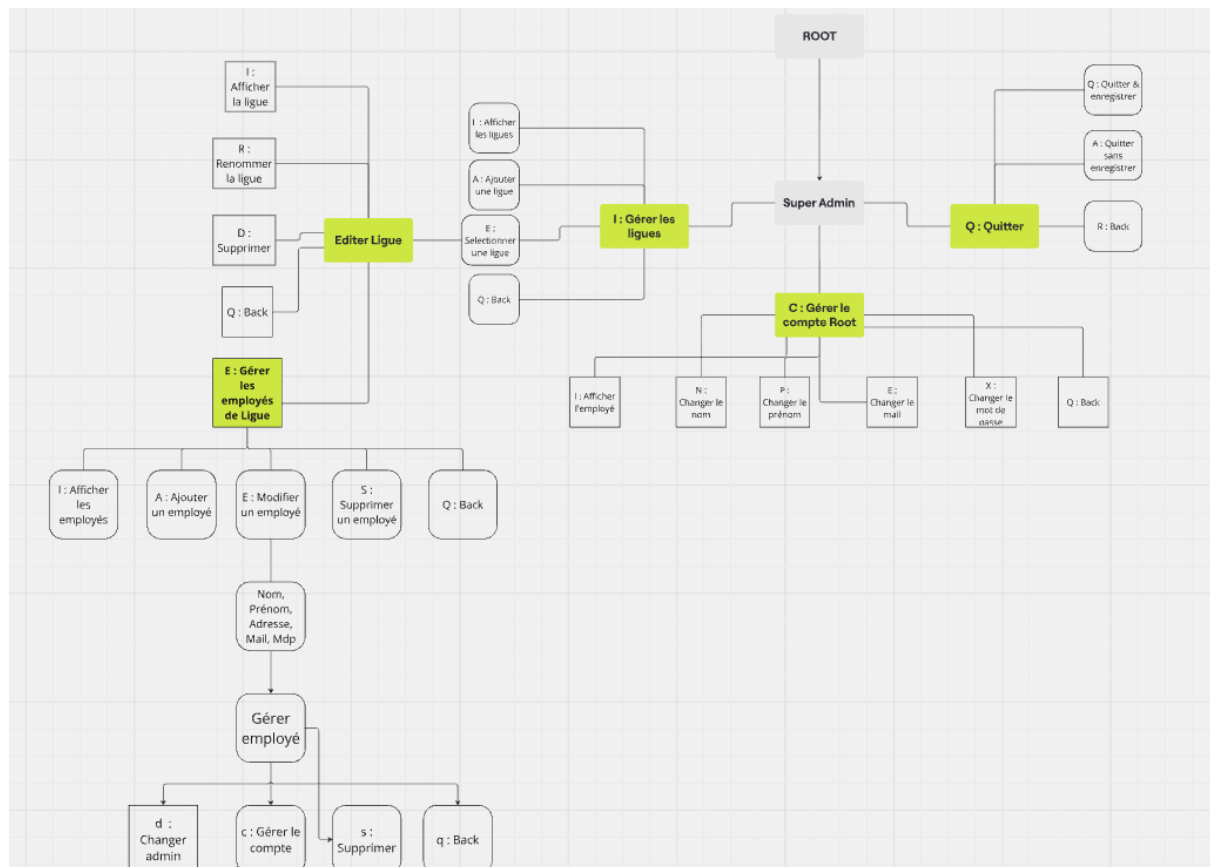
Création de la base de donnée (MCD + SCRIPT SQL) & Arbre heuristique



[Lien vers le commit du MCD](#)

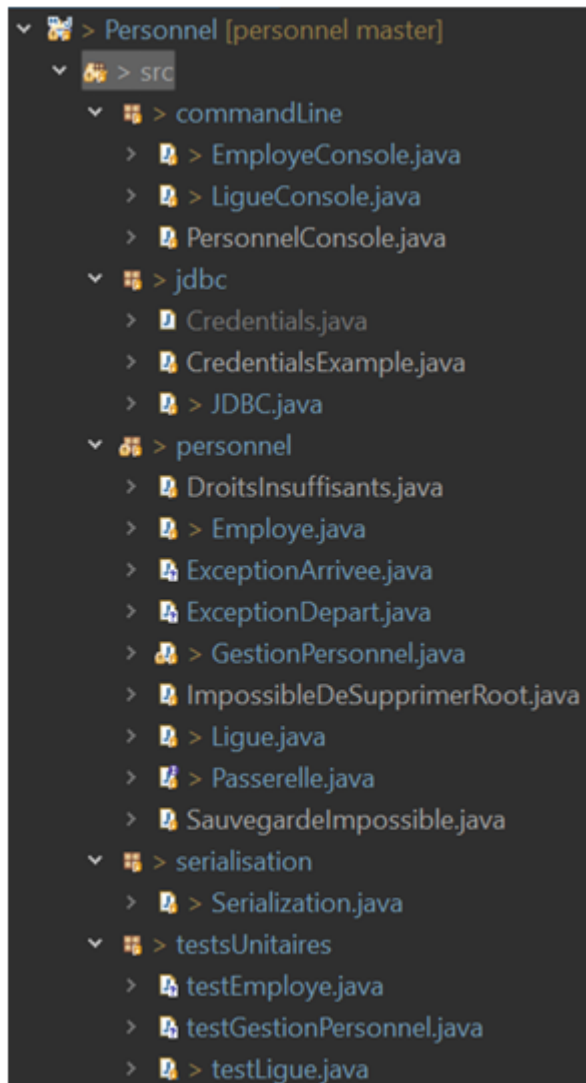
```
1  DROP TABLE if exists Ligue;
2  DROP TABLE if exists Employe;
3
4  CREATE TABLE Ligue(
5      ID_LIGUE INT,
6      NOM_L VARCHAR(),
7      PRIMARY KEY(ID_LIGUE)
8  )engine = innodb;
9
10 CREATE TABLE Employe(
11     ID_EMP INT,
12     NOM_EMP VARCHAR(),
13     PRENOM_EMP VARCHAR(),
14     MDP_EMP VARCHAR(),
15     DATE_ARRIVE DATE(),
16     DATE_DEPART DATE(),
17     ADMIN BOOLEAN,
18     MAIL_EMP VARCHAR(),
19     PRIMARY KEY(ID_EMP),
20     FOREIGN KEY(ID_LIGUE) references LIGUE(ID_LIGUE)
21 )engine = innodb;
```

[Lien vers le commit du script SQL](#)



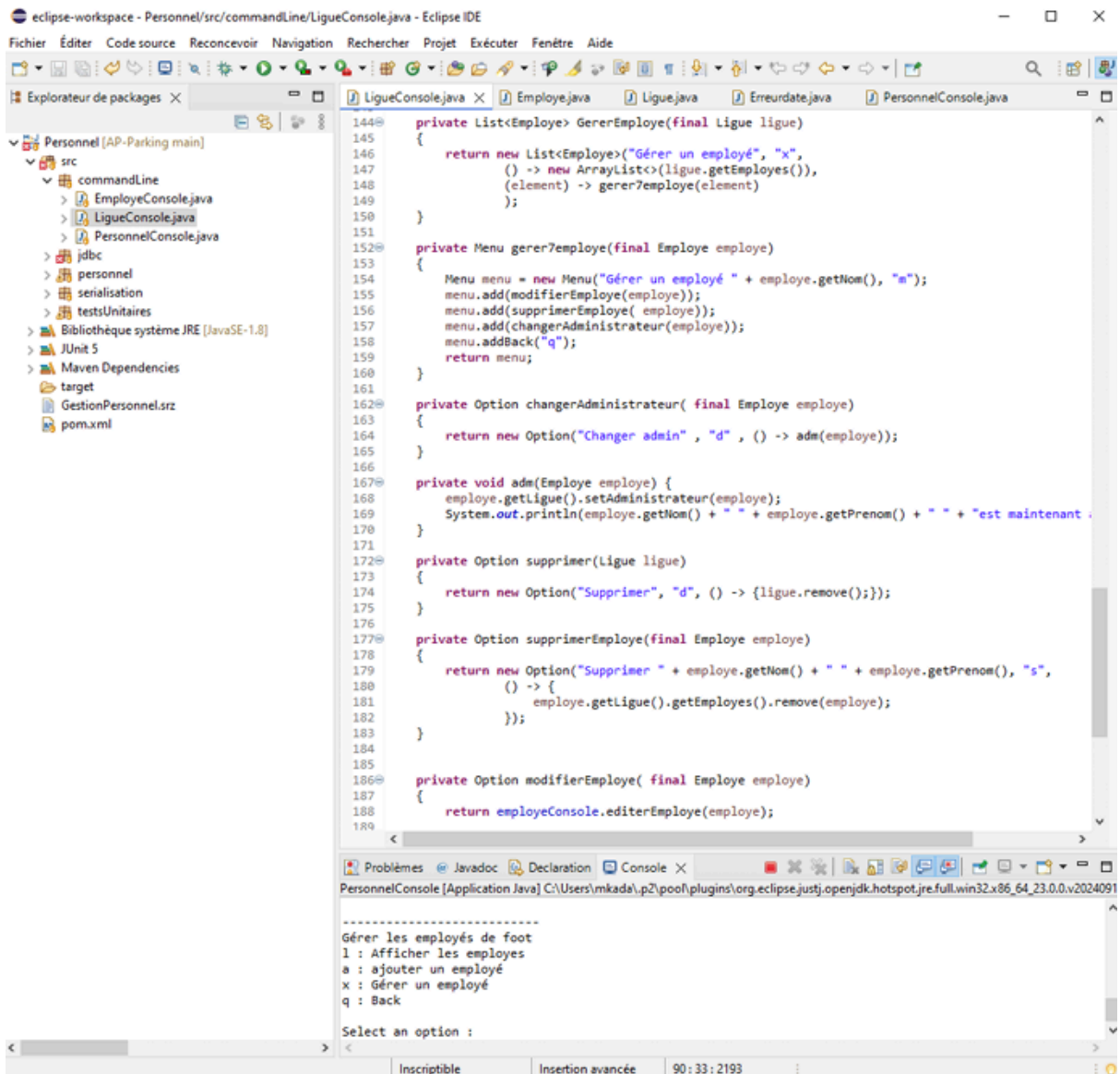
[Lien vers le commit de l'arbre heuristique](#)

2 / L'application Personnel



```
Gestion du personnel des ligues
c : Gerer le compte root
l : Gerer les ligues
q : Quitter

Select an option : |
```



Ensuite nous avons décidé [d'améliorer la prise en considération des erreurs](#), notamment concernant les dates d'arrivées et de départs, par exemple concernant le format de la date.

```
private Option ajouterEmploye(final Ligue ligue)
{
    return new Option("ajouter un employé", "a",
        () ->
        {
            // Saisie en chaîne - demande toutes les informations d'abord
            String nom = getString("nom : ");
            String prenom = getString("prenom : ");
            String mail = getString("mail : ");
            String password = getString("password : ");
            String dateArrStr = getString("date arrivée (AAAA-MM-JJ) : ");
            String dateDepStr = getString("date départ (AAAA-MM-JJ) : ");

            try {
                // Exception Format - vérifie le format des dates
                LocalDate dateArrivee = LocalDate.parse(dateArrStr);
                LocalDate dateDepart = LocalDate.parse(dateDepStr);

                // Exception ordre - vérifie l'ordre chronologique
                if (dateDepart.isBefore(dateArrivee)) {
                    throw new Erreurdate();
                }

                // Si tout est valide, crée l'employé
                ligue.addEmploye(nom, prenom, mail, password, dateArrivee, dateDepart);
                System.out.println("Employé ajouté avec succès");
            } catch (DateTimeParseException e) {
                // Gestion de l'exception de format
                System.out.println("Erreur : Format de date invalide. Utilisez le format AAAA-MM-JJ");
            } catch (Erreurdate e) {
                // Gestion de l'exception d'ordre chronologique
                System.out.println(e.getMessage());
            }
        }
    );
}
```