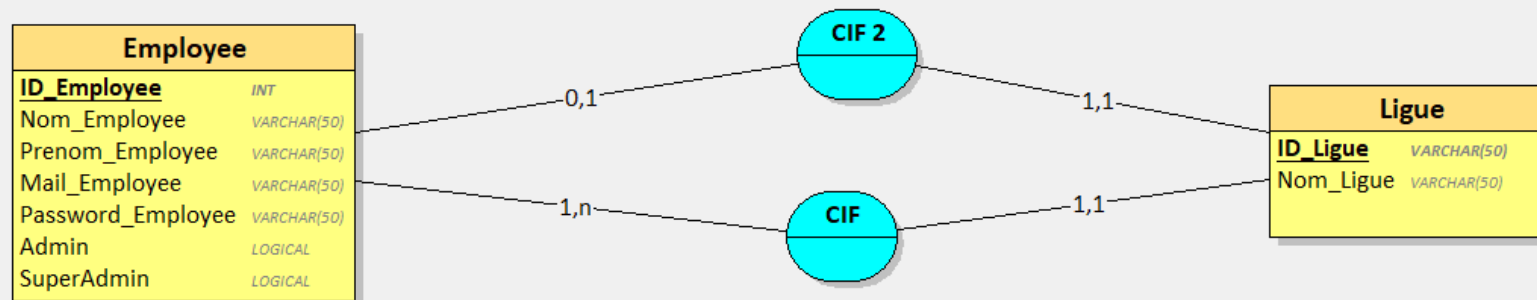


Compte Rendu Itération

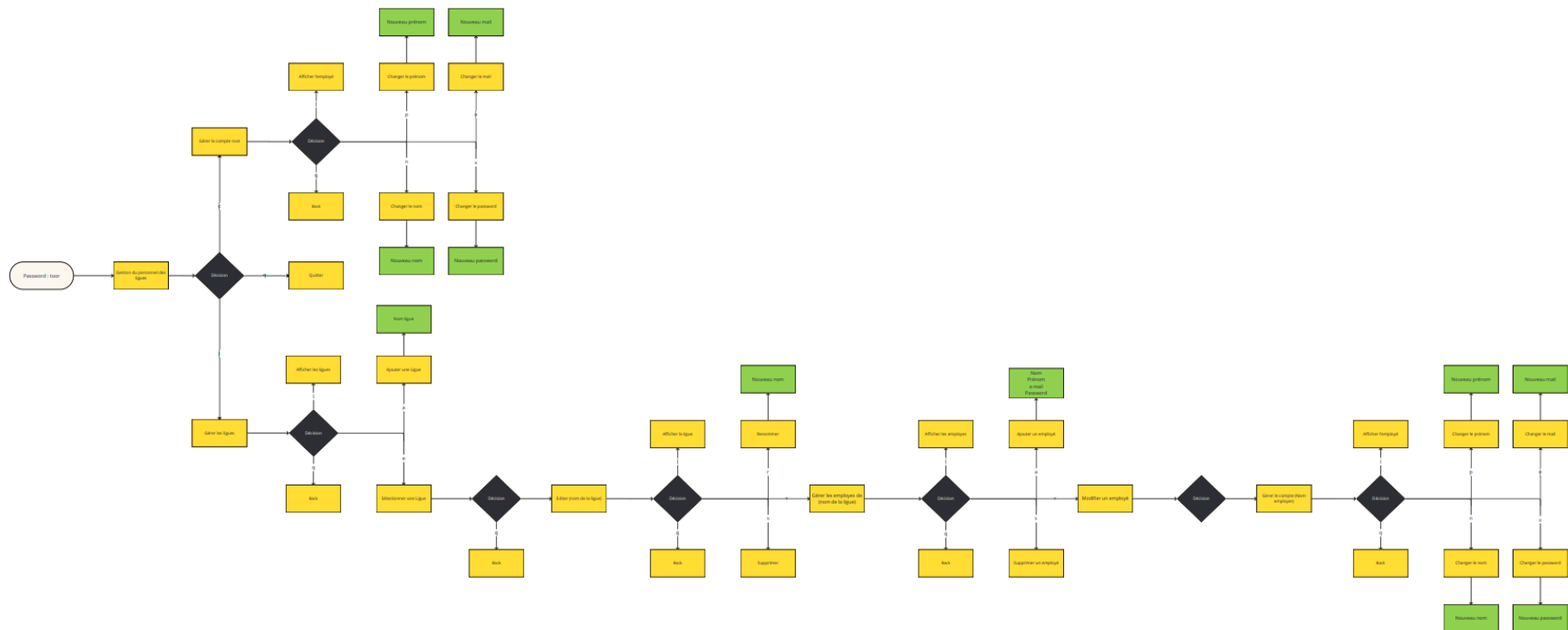
Création du MCD :

Le MCD présente deux entités principales : Employee et Ligue. L'entité Employee contient des attributs tels que l'ID, le nom, le prénom, l'e-mail, le mot de passe, et des champs pour indiquer les droits d'admin et de super-admin. L'entité Ligue comprend l'ID et le nom de la ligue. Deux relations sont définies : l'une est une relation de **1,n** entre Employee et Ligue, indiquant qu'un employé peut être associé à plusieurs ligues ; l'autre est une relation **0,1** à **1,1**, suggérant qu'une ligue peut être optionnellement liée à un employé spécifique. Ce MCD nous a permis de visualiser les entités et leurs relations dans le système, ce qui nous a facilité la conception de la maquette.



Création de la maquette :

La maquette présente le système de gestion destiné aux ligues et au personnel associé. Il propose plusieurs fonctionnalités, telles que la gestion du compte root, la modification des informations des employés (nom, prénom, e-mail, mot de passe), et la gestion des ligues (ajout, suppression, et modification du nom des ligues). Les utilisateurs peuvent naviguer à travers des options pour afficher les ligues et les employés, ajouter ou supprimer des employés, ainsi qu'éditer les détails des comptes. Chaque action est décrite avec des choix précis permettant d'effectuer différentes opérations, structurées autour de décisions claires pour la gestion des données et des utilisateurs. ([Voir GitHub](#) pour une meilleure qualité.)



Création du script des tables SQL :

Nous avons utilisé le MCD réalisé dans Itinération-1 pour réaliser ce script des tables SQL. Ce script sert à définir la structure de la base de données pour gérer le système de ligues et d'employés. Voici ses objectifs principaux :

Stocker les ligues : La table *Ligue* permet de garder les informations de chaque ligue, tels que son identifiant ou encore son nom, ce qui permet d'identifier et de nommer chaque ligue.

Stocker les employés : La table *Employe* garde les informations de chaque employé, comme leur prénom, nom email ou encore mot de passe. Leur statut admin y est aussi défini, et chaque employé est lié à une ligue grâce à la clé étrangère ID_ligue.

Gestion des relations : Une contrainte permet qu'un employé ne peut être associé à une seule et même ligue, ce qui garanti que chaque employé est toujours associé à une ligue existante.

Tout cela permet l'efficacité de la gestion des ligues, des employés et de leur relation tout en étant cohérent.

```
1  CREATE TABLE Ligue (  
2      ID_Ligue INT,  
3      nomLigue VARCHAR(100) NOT NULL,  
4      ADD CONSTRAINT PK_Ligue PRIMARY KEY (ID_Ligue)  
5  )engine = innodb;  
6  
7  CREATE TABLE Employe (  
8      ID_Employe INT,  
9      prenomEmploye VARCHAR(50) NOT NULL,  
10     nomEmploye VARCHAR(50) NOT NULL,  
11     mail VARCHAR(100) UNIQUE NOT NULL,  
12     passwd VARCHAR(50) NOT NULL,  
13     datearrv DATE NOT NULL,  
14     datedepart DATE,  
15     Admin BOOLEAN NOT NULL,  
16     ID_Ligue INT,  
17     ADD CONSTRAINT PK_Employe PRIMARY KEY (ID_Employe)  
18 )engine = innodb;  
19  
20 ALTER TABLE Employe  
21 ADD CONSTRAINT fk_Ligue_employe  
22 FOREIGN KEY (ID_Ligue)  
23 REFERENCES Ligue(ID_Ligue);
```

Lien du script : [Iteration2/CREATE TABLE.sql at main · WassimElArche/Iteration2](#)

Modification du Code en Java :

Le code LigueConsole propose une interface en ligne qui permet d'interagir avec l'utilisateur. Celui-ci permet de gérer des ligues et leurs employés. Voici les principales fonctionnalités :

Menu principal : Afficher les ligues, ajouter une ligue et sélectionner une ligue.

```
Menu menuLigues()
{
    Menu menu = new Menu("Gérer les ligues", "1");
    menu.add(afficherLigues());
    menu.add(ajouterLigue());
    menu.add(selectionnerLigue());
    menu.addBack("q");
    return menu;
}
```

Gestion des ligues : Afficher les détails d'une ligue, renommer une ligue, changer l'administrateur et supprimer une ligue.

```
Menu menuLigues()
{
    Menu menu = new Menu("Gérer les ligues", "1");
    menu.add(afficherLigues());
    menu.add(ajouterLigue());
    menu.add(selectionnerLigue());
    menu.addBack("q");
    return menu;
}
```

Gestion des employés : Afficher les employés, ajouter un employé, modifier un employé et supprimer un employé.

```
private Menu gererEmployes(Ligue ligue)
{
    Menu menu = new Menu("Gérer les employés de " + ligue.getNom(), "e");
    menu.add(afficherEmployes(ligue));
    menu.add(ajouterEmploye(ligue));
    menu.add(selectEmploye(ligue));
    menu.addBack("q");
    return menu;
}
```

Gestion des erreurs : Affichage de messages en cas d'erreurs et contrôle les formats.

Cela permet une gestion simple et complète des ligues et des employés pour l'utilisateur.

Lien du code : [personnel/Personnel/src/commandLine/LigueConsole.java at master · WassimElArche/personnel](https://github.com/WassimElArche/personnel/blob/master/src/commandLine/LigueConsole.java)