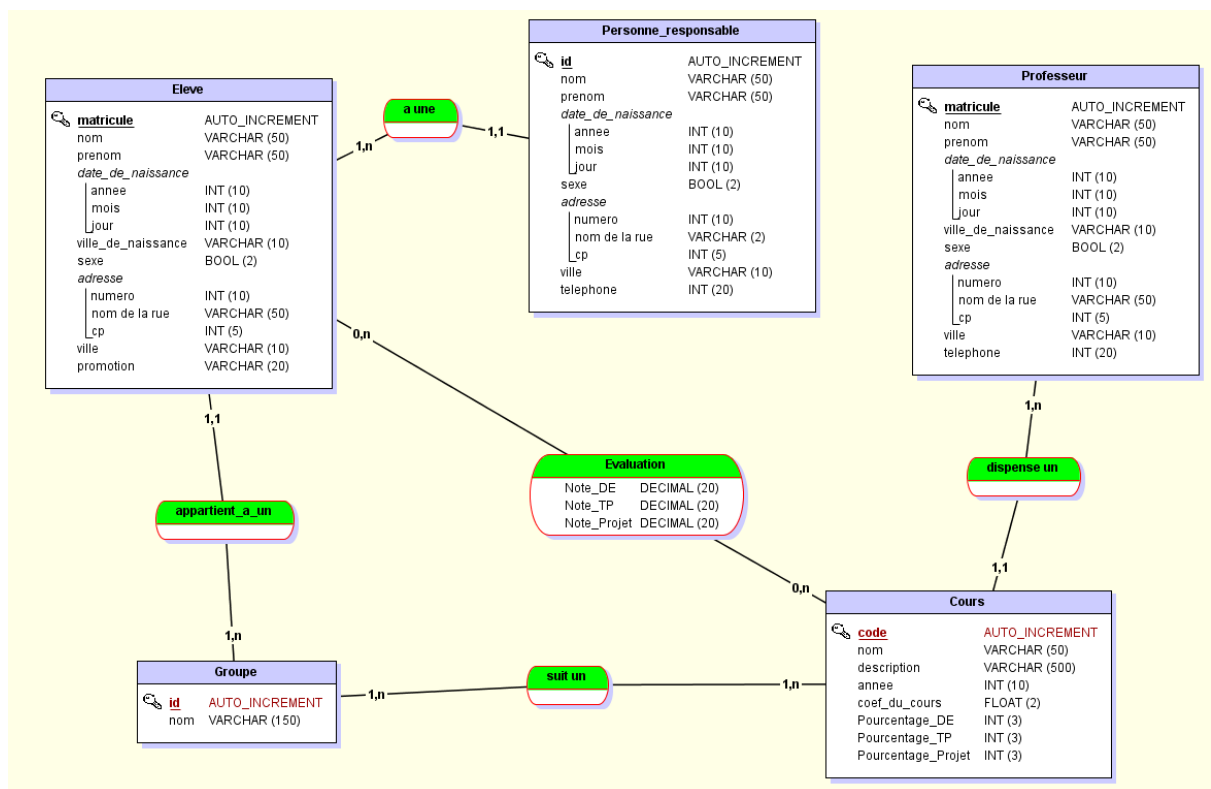


## Projet Base de données:

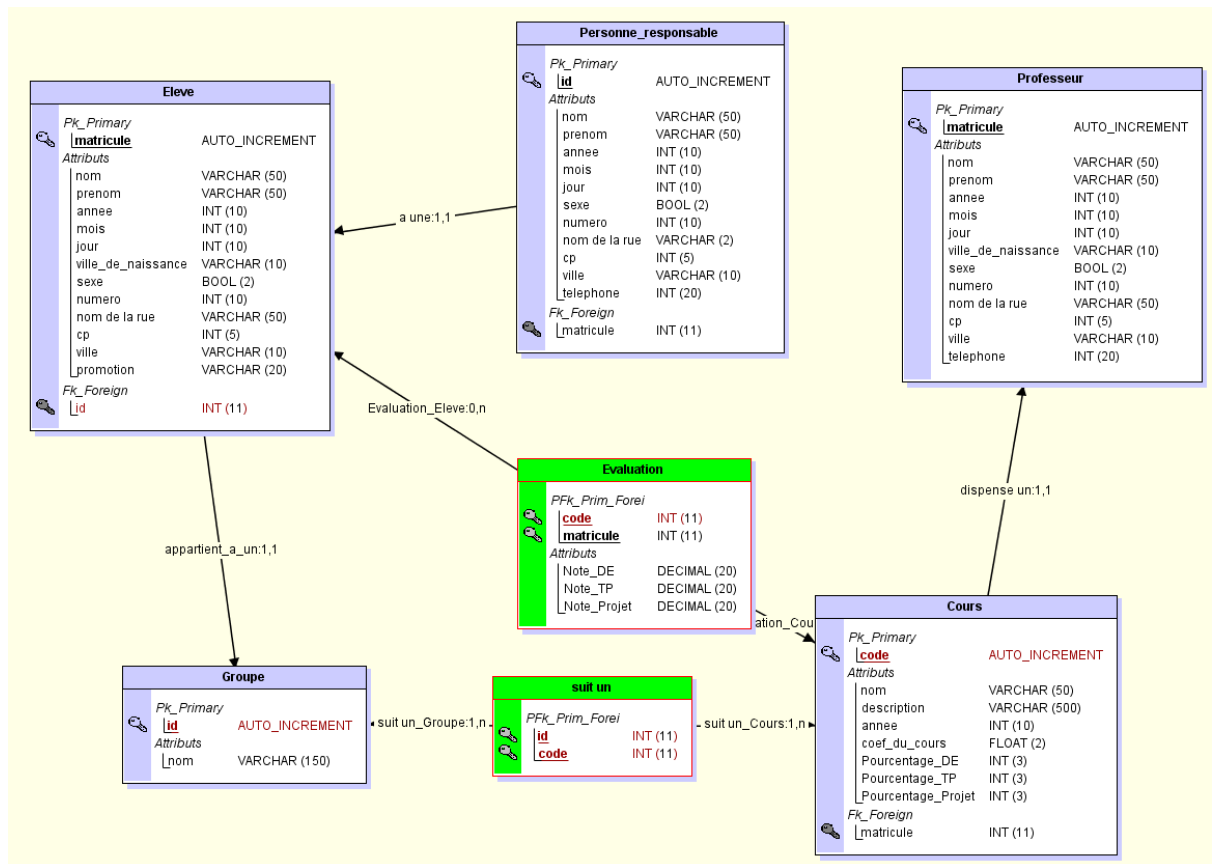
- I) Description du projet
- II) Les fonctionnalités
- III) Les difficultés rencontrées
- IV) Conclusion

### I) Description du projet

Dans le cadre du module de Base de données du semestre 6, on a eu à faire un projet de création et de gestion d'une base de données grâce à une applet JAVA. Le projet consiste à créer une base de données de gestion administrative des élèves, des cours, et des professeurs. Pour cela nous avons créé d'abord un MCD (Gestion des droits d'accès) :



Ensuite nous avons engendré le MLD (Modèle logique des données) :



Ensuite nous avons engendr  le script SQP afin de cr er les tables :

```

#-----
#      Script MySQL.
#-----

#-----
# Table: Groupe
#-----

CREATE TABLE Groupe(
    id Int Auto_increment NOT NULL ,
    nom Varchar (150)
    ,CONSTRAINT Groupe_PK PRIMARY KEY (id)
)ENGINE=InnoDB;

#-----
# Table: Eleve
#-----

CREATE TABLE Eleve(
    matricule Int Auto_increment NOT NULL ,
    nom Varchar (50) NOT NULL ,
    prenom Varchar (50) NOT NULL ,
    annee Int NOT NULL ,
    mois Int NOT NULL ,
    jour Int NOT NULL ,
    ville_de_naissance Varchar (10) NOT NULL ,
    sexe Bool NOT NULL ,
    numero Int NOT NULL ,
    nom_de_la_rue Varchar (50) NOT NULL ,
    cp Int NOT NULL ,
    ville Varchar (10) NOT NULL ,
    promotion Varchar (20) NOT NULL ,
    id Int NOT NULL
    ,CONSTRAINT Eleve_PK PRIMARY KEY (matricule)
    ,CONSTRAINT Eleve_Groupe_FK FOREIGN KEY (id) REFERENCES Groupe(id)
)ENGINE=InnoDB;

```

```
#-----  
# Table: Professeur  
#-----
```

```
CREATE TABLE Professeur(  
    matricule      Int Auto_increment NOT NULL ,  
    nom            Varchar (50) NOT NULL ,  
    prenom         Varchar (50) NOT NULL ,  
    annee          Int NOT NULL ,  
    mois           Int NOT NULL ,  
    jour           Int NOT NULL ,  
    ville_de_naissance Varchar (10) NOT NULL ,  
    sexe           Bool NOT NULL ,  
    numero         Int NOT NULL ,  
    nom_de_la_rue  Varchar (50) NOT NULL ,  
    cp             Int NOT NULL ,  
    ville          Varchar (10) NOT NULL ,  
    telephone      Int NOT NULL  
    ,CONSTRAINT Professeur_PK PRIMARY KEY (matricule)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: Cours  
#-----
```

```
CREATE TABLE Cours(  
    code           Int Auto_increment NOT NULL ,  
    nom            Varchar (50) NOT NULL ,  
    description     Varchar (500) NOT NULL ,  
    annee          Int NOT NULL ,  
    coef_du_cours  Float NOT NULL ,  
    Pourcentage_DE Int NOT NULL ,  
    Pourcentage_TP Int NOT NULL ,  
    Pourcentage_Projet Int NOT NULL ,  
    matricule      Int NOT NULL  
    ,CONSTRAINT Cours_PK PRIMARY KEY (code)  
    ,CONSTRAINT Cours_Professeur_FK FOREIGN KEY (matricule) REFERENCES Professeur(matricule)  
)ENGINE=InnoDB;
```

```

#-----
# Table: Personne_responsable
#-----

CREATE TABLE Personne_responsable(
  id          Int Auto_increment NOT NULL ,
  nom         Varchar (50) NOT NULL ,
  prenom      Varchar (50) NOT NULL ,
  annee       Int NOT NULL ,
  mois        Int NOT NULL ,
  jour        Int NOT NULL ,
  sexe        Bool NOT NULL ,
  numero      Int NOT NULL ,
  nom_de_la_rue Varchar (2) NOT NULL ,
  cp          Int NOT NULL ,
  ville       Varchar (10) NOT NULL ,
  telephone   Int NOT NULL ,
  matricule    Int NOT NULL
  ,CONSTRAINT Personne_responsable_PK PRIMARY KEY (id)

  ,CONSTRAINT Personne_responsable_Eleve_FK FOREIGN KEY (matricule) REFERENCES Eleve(matricule)
)ENGINE=InnoDB;

#-----
# Table: suit un
#-----

CREATE TABLE suit_un(
  id  Int NOT NULL ,
  code Int NOT NULL
  ,CONSTRAINT suit_un_PK PRIMARY KEY (id,code)

  ,CONSTRAINT suit_un_Groupe_FK FOREIGN KEY (id) REFERENCES Groupe(id)
  ,CONSTRAINT suit_un_Cours0_FK FOREIGN KEY (code) REFERENCES Cours(code)
)ENGINE=InnoDB;

#-----
# Table: Evaluation
#-----

CREATE TABLE Evaluation(
  code      Int NOT NULL ,
  matricule Int NOT NULL ,
  Note_DE   Decimal (20) NOT NULL ,
  Note_TP   Decimal (20) NOT NULL ,
  Note_Projet Decimal (20) NOT NULL
  ,CONSTRAINT Evaluation_PK PRIMARY KEY (code,matricule)

  ,CONSTRAINT Evaluation_Cours_FK FOREIGN KEY (code) REFERENCES Cours(code)
  ,CONSTRAINT Evaluation_Eleve0_FK FOREIGN KEY (matricule) REFERENCES Eleve(matricule)
)ENGINE=InnoDB;

```

## II) Les fonctionnalités

Les fonctionnalités essentielles à la mise en marche du projet sont :

- \*) Une interface de connexion
- \*) La mise en place de 3 niveaux de connections (Élève, Professeur et Administrateur)
- \*) La consultation par l'élève de son profil ainsi que de ses notes
- \*) La consultation par le professeur de son profil, des notes de ses élèves, la recherche d'élève en fonction du groupe ou de la promotion ainsi que la saisie et la modification des notes de ses élèves.
- \*) La suppression ainsi que l'ajout par l'administrateur de professeurs et d'élèves.

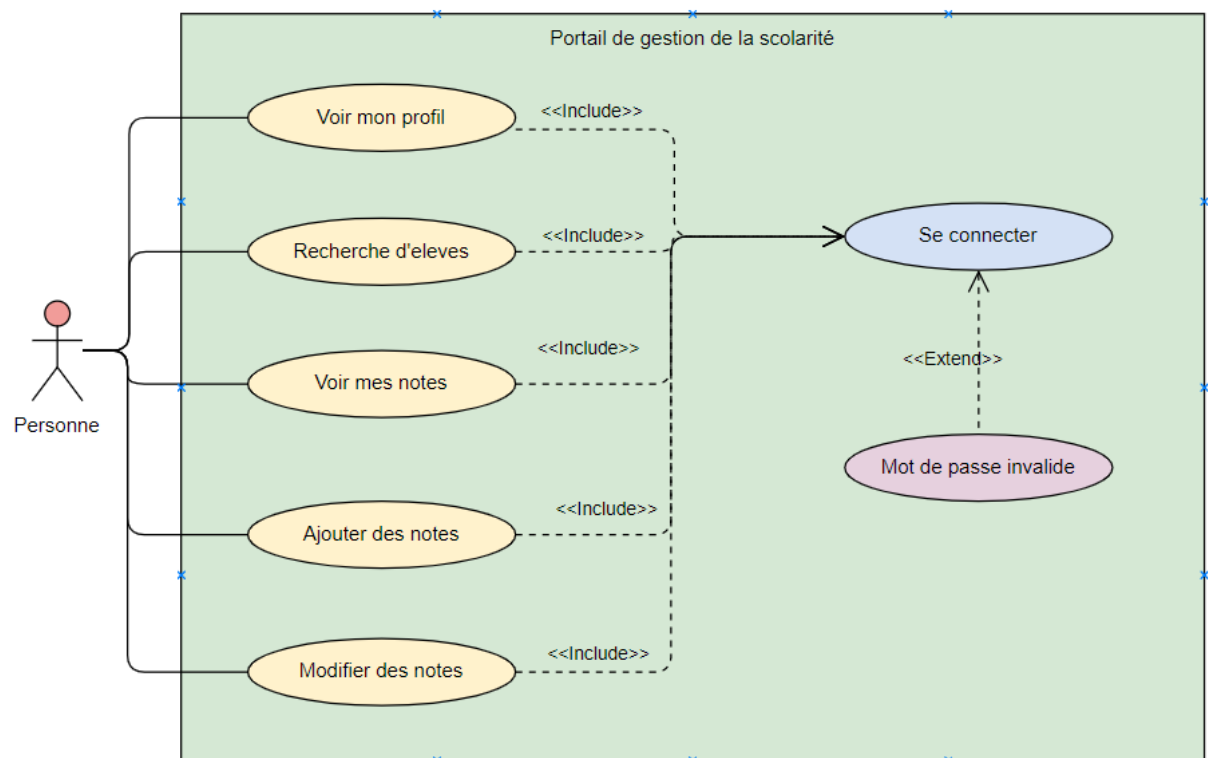
## III) Les difficultés rencontrées

Au cours du développement de notre projet, nous avons rencontrés divers problèmes, dont les plus importants sont :

- La sécurisation des saisies utilisateurs pour les recherches dans la base en évitant les injections
- La mise en place des 3 niveaux de droit pour la connexion à la base de données

Pour résumer le projet et ses fonctionnalités, voici un diagramme de cas d'utilisation du projet :

Diagramme de cas d'utilisation :



Pour les différents niveaux de connexion, le programme sait automatiquement qui se connecte en fonction de l'identifiant et le mot de passe saisi.

Enfin voici un scénario d'exécution :

L'interface de connexion :

```
run:
Bienvenue sur le nouveau portail de gestion des eleves !
Veuillez vous connectez, en saisissant votre identifiant et ensuite votre mot de passe
```

Interface pour l'administrateur :

```

run:
Bienvenue sur le nouveau portail de gestion des eleves !
Veuillez vous connectez, en saisissant votre identifiant et ensuite votre mot de passe
0
0000
Vous etes desormais connecte en tant que: Administrateur

Que voulez-vous faire ? Tapez le numero de votre choix
0)Se deconnecter !
1)Ajouter un etudiant !
2)Ajouter un prof
3)Supprimer un etudiant
4)Supprimer un prof
|

```

Interface pour l'Élève :

---

```

run:
Bienvenue sur le nouveau portail de gestion des eleves !
Veuillez vous connectez, en saisissant votre identifiant et ensuite votre mot de passe
102
123456
Vous etes desormais connecte en tant que: Eleve

Que voulez-vous faire ? Tapez le numero de votre choix
0)Se deconnecter !
1)Voir mes notes !
2)Voir mon profil
|

```

Si on choisit de voir notre profil :

```

0)Se deconnecter !
1)Voir mes notes !
2)Voir mon profil
2

Profil:

Matricule: 2   Nom: Hamdi   Prenom: Badr   Date de naissance: 6/6/1997   Ville de naissance: Paris   Sexe: homme
Adresse: 45 rue_le_marois   Code postal: 75016   Ville: Paris   Promotion: 2021

Que voulez-vous faire ? Tapez le numero de votre choix
0)Se deconnecter !
1)Voir mes notes !
2)Voir mon profil
|

```

---

Interface du Prof :

```

run:
Bienvenue sur le nouveau portail de gestion des eleves !
Veuillez vous connectez, en saisissant votre identifiant et ensuite votre mot de passe
13
12345
Vous etes desormais connecte en tant que: Professeur

Que voulez-vous faire ? Tapez le numero de votre choix
0)Se deconnecter
1)Voir les notes de mes eleves!
2)Saisir des notes
3)Modifier des notes
4)Recherche d'eleves
5)Voir mon profil
|

```

Si on décide de rechercher un groupe d'élèves alors on tape 4 et on a :

```

2)Saisir des notes
3)Modifier des notes
4)Recherche d'eleves
5)Voir mon profil
4
Que voulez-vous chercher ? Tapez le numero de votre choix !
1)Une liste d'eleves par groupe
2)Une liste d'eleves par promotion
2
Saisir le nom de la promotion que vous voulez chercher !
2021
Matricule: 2   nom: Hamdi   prenom: Badr   Promotion: 2021
Matricule: 13  nom: Jin    prenom: Edouard   Promotion: 2021
Matricule: 14  nom: Taccola prenom: Alessandro Promotion: 2021
Matricule: 19  nom: Taccola prenom: Alessandro Promotion: 2021

```

#### IV) Conclusion

En conclusion, le projet nous a appris comment faire une connexion entre la base de donnée et l'applet JAVA, manier les 2 langages JAVA et SQL. Les grandes difficultés ont été de faire la connexion au début, puis de bien vérifier que la personne se connectant est soit un élève, soit un prof ou un administrateur. Enfin, faire attention à ce que l'utilisateur ne puisse pas faire d'injection SQL dans la saisie afin d'avoir accès à des informations dont il n'a pas le droit, c'est pour cela que nous avons utilisé les requêtes préparées afin d'empêcher cette faille de se produire.