

Groupe 3 - Ony ANDRIATSAHAVOJAONA  
Océane LUYEYE LU MVUALA

### **S2.04 Exploitation d'une Base de Données**

Table représentatives de la base de donnée :

**Clés primaires en couleur rouge**

- Table Etudiant ( **Etud\_id int primary key**, Nom\_etud varchar(50) , Prenom\_etud varchar(50), Groupe varchar(50));
- Table Ressources ( **Ressource\_id varchar(50) primary key**, Nom\_res varchar(50) , Coefficient float , Ens\_id int, Heures\_res int, Cours varchar(50), Prerequis varchar(50));
- Table Enseignant ( **Ens\_id int primary key**, Nom\_ens varchar(50) , Prenom\_ens varchar(50), Ressource\_id varchar(50) );
- Table Notes ( **Etud\_id int primary key**, **Ressource\_id varchar(50) primary key**, Note float, Coefficient float);
- Table Sae ( **Sae\_id varchar(50) primary key**, Etud\_id int , Note float , Nom\_sae varchar(50), Coefficient float, Cours varchar(50), Competence varchar(50), Heures\_sae int);
- Table Responsable\_res ( **Resp\_id int primary key**, Nom\_ens varchar(50), Prenom\_ens varchar(50), Ressource\_id varchar(50), Ens\_id int);
- Table Contrôle ( **Contrôle id varchar(50) primary key**, Ressource id varchar(50), Coefficient float, Note float, Etud\_id int foreign key (Etud id) references Etudiant (Etud id));

## Script relatifs à la création des tables de notre base de donnée :

jupyter SAE\_BDDS2 Dernière Sauvegarde : il y a une heure (auto-sauvegardé) Se déconnecter

Fichier Édition Affichage Insérer Cellule Noyau Widgets Aide Fiabilité Python 3

Entrée [1]: `%reload_ext sql`

Entrée [2]: `from sqlalchemy import create_engine`

Entrée [4]: `%sql postgresql://oceane:LunedeTerre@localhost/postgres`

Entrée [6]: `%sql`  
`CREATE TABLE Etudiant (Etud_id int PRIMARY KEY,`  
`Nom_etud varchar(50),`  
`Prenom_etud varchar(50),`  
`Groupe varchar(50));`

\* postgresql://oceane:\*\*\*@localhost/postgres  
Done.

Entrée [51]: `%sql`  
`CREATE TABLE Ressources (Ressource_id varchar(50) primary key,`  
`Nom_res varchar(50),`  
`Coefficient float,`  
`Ens_id int,`  
`Heures_res int,`  
`Cursus varchar(50),`  
`Prerequis varchar,`  
`foreign key (Ens_id) references Enseignant (Ens_id));`

\* postgresql://oceane:\*\*\*@localhost/postgres

Entrée [50]: `%sql`  
`CREATE TABLE Enseignant( Ens_id int primary key,`  
`Nom_ens varchar(50),`  
`Prenom_ens varchar(50),`  
`Ressource_id varchar(50),`  
`foreign key (Ressource_id) references Ressources(Ressource_id)`  
`);`

\* postgresql://oceane:\*\*\*@localhost/postgres

Entrée [38]: `%sql`  
`CREATE TABLE Notes ( Etud_id int primary key,`  
`Ressource_id varchar(50) primary key,`  
`Note float,`  
`Coefficient float,`  
`foreign key (Ressource_id) references Ressources (Ressource_id),`  
`foreign key (Etud_id) references Etudiant (Etud_id));`

\* postgresql://oceane:\*\*\*@localhost/postgres

Entrée [41]: `%%sql`

```
CREATE TABLE SAE ( Sae_id varchar(50) primary key,
                    Etud_id int ,
                    Note float ,
                    Nom_sae varchar(50),
                    Coefficient float,
                    Cursus varchar(50),
                    Heures_sae int,
                    Compétence varchar(50),
                    foreign key (Etud_id) references Etudiant (Etud_id));
```

`* postgresql://oceane:***@localhost/postgres`  
Done.

Entrée [53]: `%%sql`

```
CREATE Table Responsable_res ( Resp_id int primary key,
                               Nom_ens varchar(50),
                               Prenom_ens varchar(50),
                               Ressource_id varchar(50),
                               Ens_id int,
                               foreign key (Ressource_id) references Ressources (Ressource_id),
                               foreign key (Ens_id) references Enseignant (Ens_id));
```

`* postgresql://oceane:***@localhost/postgres`  
Done.

Entrée [55]: `%%sql`

```
CREATE Table Contrôle (Contrôle_id varchar(50) primary key,
                       Ressource_id varchar(50),
                       Coefficient float,
                       Note float,
                       Etud_id int,
                       foreign key (Etud_id) references Etudiant (Etud_id) );
```

`* postgresql://oceane:***@localhost/postgres`

### Script relatif à l'insertion de donnée dans une table de notre base de données:

Script pour insérer des données au seins de nos tables

Entrée [25]: `%%sql`

```
insert into Etudiant values(12103382,'Luyeye Lu Mvuala', 'Oceane','Pegasus');
insert into Etudiant values(12103123,'ANDRIATSAHAVOJOANA','Ony','Pegasus');
```

`* postgresql://oceane:***@localhost/postgres`  
1 rows affected.  
1 rows affected.

### Description des tables :

- Table Etudiant : Cette table contient toutes les informations concernant l'étudiant, Nom , Prénom , Groupe et Numéro d'étudiant qui fait office de clé primaire car le numéro d'étudiant permet une identification unique de chaque étudiant.
- Table Ressources : Cette table contient toutes les informations concernant une ressources, l'attribut Ressource\_id fait office de clé primaire. L'attribut cursus permet de déterminer si ce sont des ressources appartenant au 1er ou second semestre.
- Table Enseignant : La table Enseignant possède toutes les informations concernant les enseignants. L'attribut Ens\_id représente l'identifiant des enseignants donc fait office de clé primaire de la table Enseignant. L'attribut ressource\_id clé primaire de la **Table Ressources** permet de faire la liaison entre la **table Enseignant** et **Ressources**.
- Table Notes : La table Notes contient toutes les données concernant les notes des étudiants du BUT informatique. Elle contient également l'attribut Etud\_id qui représente la clé primaire de la table étudiant. Cet attribut est indispensable dans la **table Notes** pour répertorier au mieux les notes appartenant à un étudiant. De plus, elle possède l'attribut Ressource\_id clé primaire de la **table Ressources** car chaque note obtenue est associée à une ressource.
- Table Sae : La table SAE contient toutes les données concernant les sae, projet effectué au cours des deux semestres. L'attribut Compétence représente la compétence principale acquise dans une SAE.
- Table Responsable\_Res : La table Responsable\_Res répertorie tous les responsables de ressources au sein du BUT informatique. Elle contient pour ce faire Ens\_id l'identifiant des enseignants mais également Ressource\_id pour répertorier les responsables de ressource à la ressource à laquelle ils sont assignés.
- Table Contrôle : La table Contrôle contient toutes les informations relatives à un contrôle. Celle-ci possède une clé primaire qui est Controle\_id. Elle possède également l'attribut Etud\_id clé primaire de la **table Etudiant** afin d'établir une relation entre les deux tables dans le but de répertorier au mieux les contrôles appartenant à chaque étudiant.

#### CAHIER DES CHARGES :

Domaine : Département informatique de l'I.U.T. de Villetaneuse

Contexte: Dans le cadre de la gestion des notes des étudiants au sein du BUT informatique, il nous a été demandé de créer une base de données permettant de stocker à la fois toutes les notes des étudiants, appartenant à différents groupes, mais également toutes les informations relatives à l'étudiant et à ces notes (ressources étudiées tout au long de l'année, notes obtenues suite aux différents examens, contrôles effectués...)

Objectif: Établir une base de données afin de gérer au mieux les notes des étudiants au sein du BUT informatique à l'aide de règles de gestion et d'accès aux données ainsi que de leur mise en œuvre par des procédures stockées, des vues ou des vues matérialisées. Notre base de données vise à être utilisée au sein d'une IUT, spécifiquement par des professeurs. Ces professeurs seront en mesure d'afficher certaines notes et d'en interdire l'accès à d'autres. L'objectif principal du projet réside dans la gestion des notes appartenant à un étudiant. Pour ce faire, il va falloir organiser la base de données de façon à ce que les requêtes implémentées soient efficaces et puissent être d'une grande utilité pour les futurs utilisateurs.

Besoin fonctionnels :

- La base de données doit stocker toutes les données relatives à l'étudiant
- Les tables constituant la base de données doivent respecter les 3 Formes Normales
- Chaque tuple contenu dans chaque table de la base de données doit contenir des données atomiques
- Chaque table appartenant à la base de données doit contenir au moins une clé primaire et au moins une clé étrangère.
- Les données appartenant à la base doivent être visibles et lisibles
- La base de données doit être en mesure de retourner des résultats simples et lisibles de tous
- Les tables appartenant à la base de données doivent être liées entre elles
- Base de données accessible à plusieurs clients

Les délais de livraison :

- Le projet débute le 8 Mars et la première partie est à rendre le 19 avril 2022.

La procédure de réception :

- La base de données doit être réalisée sur le poste local des étudiants via le système de gestion de base de données postgresql (SGBD).
- Le rendu se fait par mail au référent pédagogique.

### Contraintes:

- Doit contenir un nombre élevé de données classées en fonction d'un ordre prédéfinis
- Respecter les 3FN (formes normales)
- Les notes des étudiants ne doivent pas être accessibles par d'autres étudiants
- La base de données ne doit être accessible qu'à un panel de personnes déterminé (dans ce cas précis les professeurs).

### Exigences:

- Doit garantir la cohérence de données
- Doit garantir la sécurité des données
- Le temps de réponses doit être optimal
- Doit permettre des accès simultanés à plusieurs utilisateurs
- Doit permettre l'accès aux enseignants ainsi qu'aux élèves aux notes
- Doit permettre aux enseignants de pouvoir changer les notes
- Ne doit pas permettre aux élèves de pouvoir changer les notes

### Résultats attendus :

- Être en mesure de visualiser les notes des étudiants en fonction des différentes matières
- Être en mesure de prendre connaissance des différents coefficients de chaque contrôle

### Technique :

- Cohérence de données au sein de la base
- Doit permettre l'intégration d'autres logiciel visant à améliorer la base

### Performance :

- Extraction rapide des données

### Ergonomie :

- Accessibilité simple des données
- Doit être utilisable par tout type de personnes

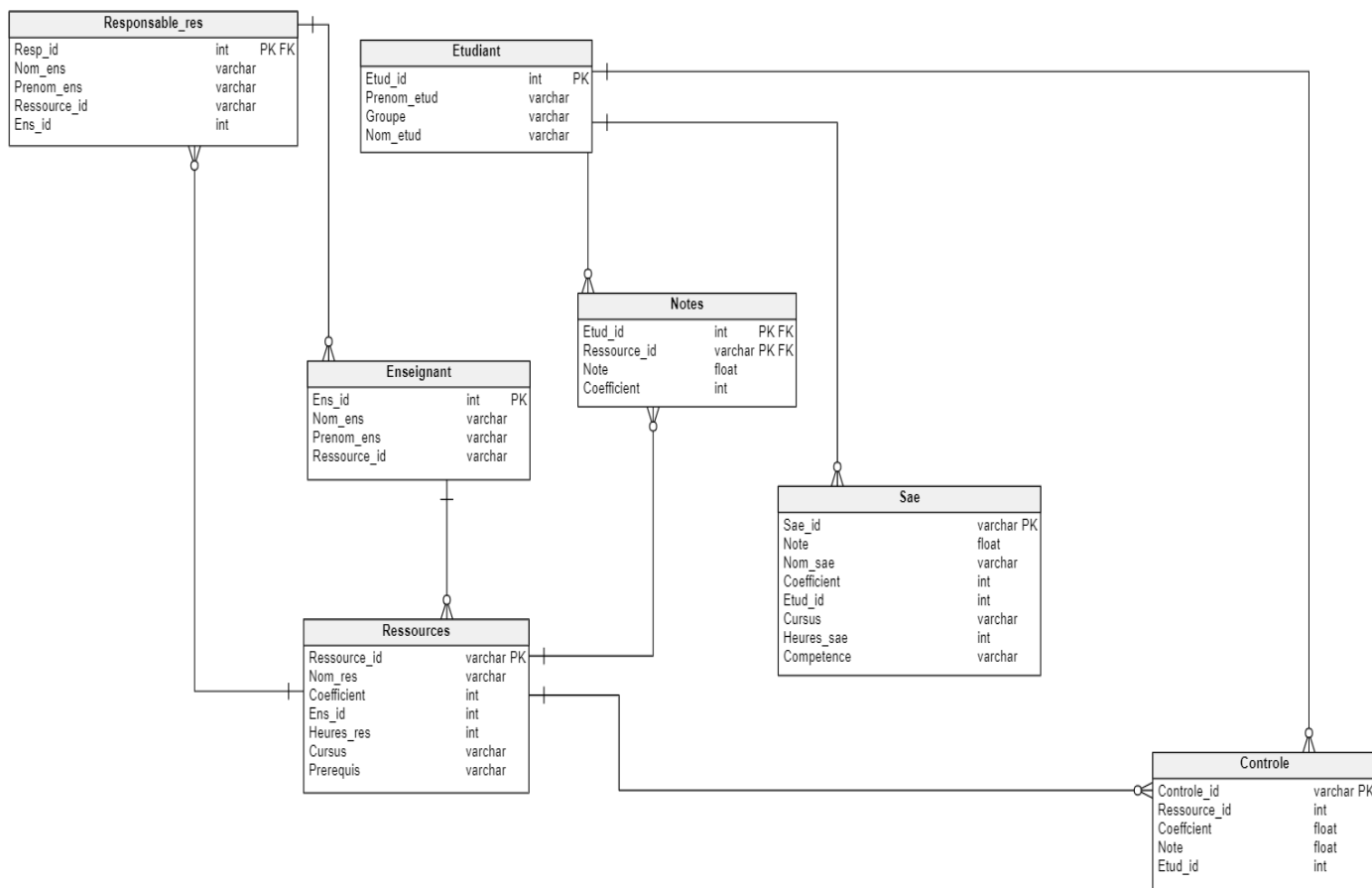
### Sécurité :

- Doit garantir la sécurité des données
- Gérer le risque de piratage
- Gérer la corruption et l'incohérence des données

### Conditions de maintenance de la base de donnée :

- Update
- Delete
- Insert

### Question N\*2 : Étudier un modèle de données et réaliser une Base de Données à partir de ce modèle



**PK** : Primary Key - clé primaire

### Question N\*3 : Définir des règles de gestion de ces données et leurs mises en œuvre par des procédures stockées.

- Les enseignants doivent être en mesure de modifier les notes des étudiants



- Un étudiant doit avoir accès à ces notes mais ne peut pas avoir accès aux notes des autres étudiants
- Les enseignants doivent avoir accès aux notes des étudiants de leur groupe
- Les enseignants ont la capacité de modifier les notes des étudiants du groupe auquel ils sont assignés pour une ressource
- Les notes ne doivent pas être modifiées par les élèves
- Le responsable d'une ressource a accès à la totalité des notes des étudiants de sa ressource respective
- Le responsable d'une ressource a également la capacité de modifier les notes des étudiants de sa ressource respective