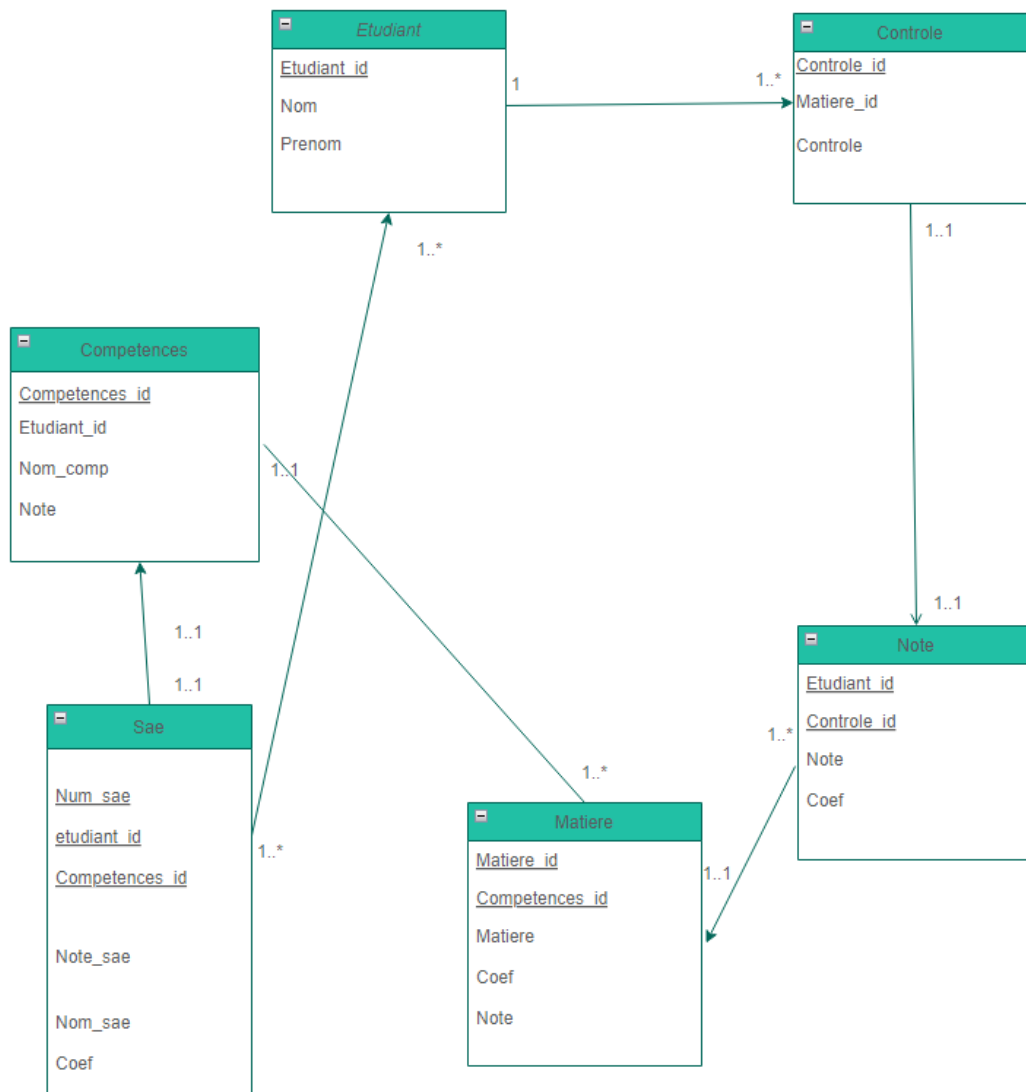


## SCHÉMA DU MODELE DE DONNÉES



Etudiant 1..1 -----1,\* Contrôle

Pour un Etudiant on peut avoir un ou plusieurs controles

Note 1..1 ----- 1.1 Contrôle

Pour une note on peut avoir un contrôle ( 1 contrôle=1note)

Matière 1.1-----1,\* Notes

Pour une matière on peut avoir plusieurs Notes

Compétences 1..1-----1,\*Matière

Pour une compétence on peut avoir plusieurs Matières

Compétences 1..1-----1.1 SAE

Pour une compétence on peut avoir une seule SAE

SAE 1..\*----- 1..\* ETUDIANT

Pour une Sae on peut avoir un ou plusieurs étudiants ( et inversement)

Voici le script de la création de notre table :

On a la table Etudiant qui représente les étudiants, elle contient un id pour les différencier, le nom de l'étudiant et le prénom.

On a La table contrôle contient son id, l'id de la matière a laquelle il est associé ainsi que l'attribut contrôle qui est de type varchar qui représente son nom.

On a la table Note qui contient l'id de l'étudiant, l'id du contrôle, le coef du contrôle et la note obtenue par l'étudiant dans ce contrôle.

On a la table Compétences qui contient l'id de la compétence, le nom de la compétence, la note obtenue dans la compétence, ainsi que l'id de l'étudiant.

On a la table Matière qui contient l'id de la matière, l'id de la compétence, le nom de la matière, le coef de la matière, et la note obtenue dans la matière (Par ailleurs nous avons la clef étrangère competences\_id car une matière peut avoir plusieurs compétences et inversement) .

On a la table Sae qui contient le num de la sae, l'id de l'étudiant, l'id de la compétence, le nom de la sae, la note de la sae ainsi que le coef (Par ailleurs nous avons la clef étrangère Etudiant\_id car une SAE peut avoir plusieurs étudiants et inversement) .

```
1 CREATE TABLE Etudiant
2 (
3     Etudiant_id serial primary key,
4     Nom varchar(50),
5     Prenom varchar(50)
6 );
7
8 CREATE TABLE Controle
9 (
10     Controle_id serial primary key,
11     Matiere_id integer(20) references Matiere(Matiere_id),
12     Controle varchar(50)
13 );
14
15 CREATE TABLE Notes
16 (
17     Etudiant_id integer references Etudiant( Etudiant_id ),
18     Controle_id integer references Controle( Controle_id ),
19     Note float(10),
20     Coef integer,
21     primary key(Etudiant_id, Controle_id)
22 );
23
24 CREATE TABLE Competences
25 (
26     Competences_id serial,
27     Nom_comp varchar(30),
28     note_comp float(15),
29     Etudiant_id integer references Etudiant(Etudiant_id),
30     primary key(Competences_id, Etudiant_id)
31 );
32
33 CREATE TABLE Matiere
34 (
35     Matiere_id serial,
36     Competences_id integer references Competences(Competences_id),
37     Matiere varchar(50),
38     Coef integer,
39     note float(5),
40     primary key(Matiere_id, Competences_id)
41 );
42
43 CREATE TABLE SAE(
44     Num_sae integer,
45     Etudiant_id integer references Etudiant(Etudiant_id),
46     Competences_id integer references Competences(Competences_id),
47     Nom_sae varchar(20),
48     Note_sae float(10),
49     Coef integer(5),
50     primary key(Num_sae, Competences_id, Etudiant_id)
51 );
```

La création des tables à bien été exécutée sans erreurs

```
✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0132 seconde(s))

CREATE TABLE Etudiant ( Etudiant_id serial primary key, Nom varchar(50), Prenom varchar(50) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0025 seconde(s))

CREATE TABLE Controle ( Controle_id serial primary key, Matiere_id integer(20) references Matiere(Matiere_id), Controle varchar(50) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0024 seconde(s))

CREATE TABLE Notes ( Etudiant_id integer references Etudiant( Etudiant_id ), Controle_id integer references Controle( Controle_id ), Note float(10), Coef integer, primary key(Etudiant_id, Controle_id) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0026 seconde(s))

CREATE TABLE Competences ( Competences_id serial, Nom_comp varchar(30), note_comp float(15), Etudiant_id integer references Etudiant(Etudiant_id), primary key(Competences_id,Etudiant_id) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0028 seconde(s))

CREATE TABLE Matiere ( Matiere_id serial, Competences_id integer references Competences(Competences_id), Matiere varchar(50), Coef integer, note float(5), primary key(Matiere_id, Competences_id) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0026 seconde(s))

CREATE TABLE SAE( Num_sae integer, Etudiant_id integer references Etudiant(Etudiant_id), Competences_id integer references Competences(Competences_id), Nom_sae varchar(20), Note_sae float(10), Coef integer(5), primary key(Num_sae,Competences_id,Etudiant_id) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]
```

## Partie 2 :

### -- Relevé des moyennes des différents étudiants

```
CREATE FUNCTION etud_moy( in etudiant_id int,out decimal(4,2))
```

```
AS $$
```

```
select avg(note)::decimal(4,2) as moyenne
```

```
from etudiant
```

```
where etudiant=Etudiant_id;
```

```
$$ language SQL;
```

### -- Relever les notes des etudiants avec des notes de SAE supérieur à 10

```
CREATE VIEW sae_note AS
```

```
SELECT s.Note_sae,e.nom;
```

```
FROM SAE s join etudiant e on e.Etudiant_id=s.Etudiant_id;
```

```
WHERE s.Note_sae > 10;
```

--chaque etudiant peut consulter ses données

```
CREATE or REPLACE function my_donnees(out nom varchar, out prenom varchar, out Etudiant_id int)
```

```
AS $$
```

```
BEGIN
```

```
    select enom,eprenom,eEtudiant_id
```

```
    into nom, prenom, id
```

```
    from employe
```

```
    WHERE session_user=lower(enom)::name;
```

```
END;
```

```
$$
```

-- Relever les notes par compétences

```
CREATE VIEW n_comp AS
```

```
select enom_comp(competence(nom_comp)).tnote
```

```
FROM(select distinct enom_comp
```

```
from competences)
```

Partie 3:

Permettre à l'étudiant de consulter ses propres notes pour chaque contrôle avec les coefficients'

```
CREATE view etudiant_notes AS
```

```
SELECT c.Nom_comp, m.Matiere, co.Controle, n.Coef, n.Note
```

```
FROM competences c JOIN etudiant e on c.Etudiant_id = e.Etudiant_id
```

```
JOIN matiere m on c.Competences_id = m.Competences_id
```

```
JOIN controle co on co.Matiere_id = m.Matiere_id
```

```
JOIN notes n on n.Controle_id = co.Controle_id
```

```
WHERE e.Etudiant_id = CURRENT_USER;
```

```
grant select on etudiant_notes to utilisateur
```

Permettre à l'étudiant de consulter quelle sont toutes les SAE à laquel il est affilié avec le coef et la note de chaque sae'

```
CREATE view sae_etudiant AS  
SELECT s.Nom_sae, s.Coeff, s.Note_sae  
FROM sae s join etudiant e ON e.Etudiant_id = s.Etudiant_id  
WHERE e.Etudiant_id = CURRENT_USER;
```

```
grant select on sae_etudiant to utilisateur
```

Permettre à l'étudiant de savoir quels sont les contrôles qu'il a passés

```
CREATE VIEW controle_passe AS  
SELECT co.Controle  
FROM competences c JOIN etudiant e on c.Etudiant_id = e.Etudiant_id  
JOIN matiere m on c.Competences_id = m.Competences_id  
JOIN controle co on co.Matiere_id = m.Matiere_id  
JOIN notes n on n.Controle_id = co.Controle_id  
WHERE e.Etudiant_id = CURRENT_USER
```

```
grant select on controle_passe to utilisateur
```

Permettre à l'étudiant de voir la moyenne de la classe d'un contrôle

```
CREATE FUNCTION MoyControle( out Controle varchar, out Note decimal(4,2))  
    RETURNS setof record  
as  
$$
```

```
Select c.Controle, AVG(n.Note)
FROM Controle c, Notes n, Etudiant e
WHERE c.Controle_id = n.Controle_id
AND n.Etudiant_id = e.Etudiant_id
AND e.Nom = session_user
GROUP By (c.Controle)
$$ language SQL
SECURITY DEFINER;
```