Création BDD

TC => compétences, parcours, ressource

parcours => groupes TD, compétences

groupes TD => groupes TP

compétences => niveau

BUT => semestres, PN

PN => TC, parcours

semestres => UE

UE => compétences, saé, ressources

saé => ressource

UE => saé

UE => coefRx

groupe TD, ressource => enseignant

groupe TP, ressource => enseignant

ue => activité => compétence

ensemble groupe td encadré par enseignants = groupes td parcours correspondant

| Ressource |
| --- |
| libelléR |

| saé |
| --- |
| libelléS  heures autonomie |

| heures |
| --- |
| heures tp  heures td |

Explication des multiplicités entre UE et ressources :

une UE est composé de une ou plusieurs ressources c’est pour cela que l’on retrouve la multiplicité de 1.\* du côté de ressources et une ressource peut appartenir à aucune ou plusieurs UE c’est pour cela qu’on retrouve la multiplicité 0.\* du côté de la classe UE.

Explication des multiplicités entre UE et Parcours :

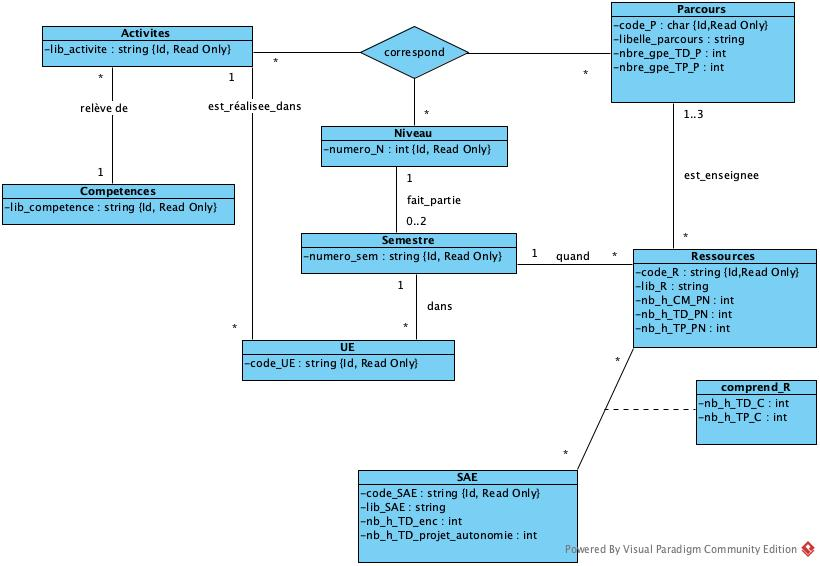
Une UE appartient à aucun, un ou plusieurs parcours c’est pour cela que nous avons décidé de mettre la multiplicité 0.\* du côté de la classe Parcours. Un Parcours contient aucune ou plusieurs UE, nous avons donc mis la multiplicité 0.\* du côté de la classe UE

Nous avons représenté le niveau de compétence dans une classe association entre Compétence et Parcours, car ce niveau dépend de la compétence à laquelle il est associé, mais aussi du parcours choisi. En effet, les compétences ne vont pas toutes jusqu’au même niveau dans tous les parcours (par exemple la compétence 5 va jusqu’au niveau 2 dans le parcours A, mais va jusqu’au niveau 3 dans le parcours C).

Les heures\_SAE s’obtiennent grâce à une classe association entre Ressources et SAE. Ce sont des heures rattachées à une SAE et elles sont complémentaires aux heures de chaque ressource du Programme National. L’information sur les heures de ressources, elles, sont directement dans la classe ressource, car c’est la manière la plus simple de les représenter, et car on ne pourrait pas créer de classe uniquement pour les heures (pas de clé primaire, à moins d’utiliser un id\_heures, ce qui n’est pas pertinent).

Pour montrer que l’activité d’une compétence appartenant à un semestre correspond à une UE, nous avons décidé de créer une association ternaire entre les classes Compétence, Activité et Semestre que nous avons nommé Correspond. Cette association permet ensuite d’obtenir la classe UE (en tant que classe association).

Nous avons opté pour cette solution car c’est la façon la plus simple en ayant les informations des classes Activité, Compétence et Semestre de faire correspondre cette classe UE.



CREATE TABLE \_parcours

(

code\_P VARCHAR(50),

libelle\_parcours VARCHAR(50),

nbre\_gpe\_TD\_P INTEGER,

nbre\_gpe\_TP\_P INTEGER

CONSTRAINT \_parcours\_pk PRIMARY KEY (code\_P)

);

CREATE TABLE \_ressources

(

code\_R VARCHAR(50),

lib\_R VARCHAR(50),

nb\_h\_CM\_PN INTEGER,

nb\_h\_TD\_PN INTEGER,

nb\_h\_TP\_PN INTEGER

CONSTRAINT \_ressources\_pk PRIMARY KEY (code\_R)

);

CREATE TABLE \_est\_enseignee

(

code\_R VARCHAR(50),

code\_P VARCHAR(50)

CONSTRAINT \_est\_enseignee\_pk

PRIMARY KEY (code\_R, code\_P)

CONSTRAINT \_est\_enseignee\_fk1 FOREIGN KEY (code\_R)

REFERENCES \_ressource(code\_R)

CONSTRAINT \_est\_enseignee\_fk2 FOREIGN KEY (code\_P)

REFERENCES \_parcours(code\_P)

);

CREATE TABLE \_sae

(

code\_sae VARCHAR(50),

lib\_sae VARCHAR(50),

nb\_h\_TD\_enc INTEGER,

nb\_h\_projet\_autonomie INTEGER

CONSTRAINT \_sae\_pk

PRIMARY KEY (code\_sae)

CREATE TABLE \_comprend\_r

(

code\_r VARCHAR(50),

code\_sae VARCHAR(50),

nb\_h\_td\_c INTEGER,

nb\_h\_tp\_c INTEGER,

CONSTRAINT \_comprend\_r\_pk

PRIMARY KEY (code\_e, code\_sae),

CONSTRAINT \_comprend\_r\_fk1 FOREIGN KEY (code\_r)

REFERENCES \_ressource(code\_R),

CONSTRAINT \_comprend\_r\_fk2 FOREIGN KEY (code\_sae)

REFERCENCES \_sae (code\_sae)

);

CREATE TABLE \_semestre

(

numero\_sem VARCHAR(50)

CONSTRAINT \_semestre\_pk

PRIMARY KEY (numero\_sem)

);

CREATE TABLE \_competences

(

lib\_competences VARCHAR(50),

CONSTRAINT \_competences\_pk

PRIMARY KEY (lib\_competences)

);

CREATE TABLE \_activites(

lib\_activite VARCHAR(50),

lib\_competence : VARCHAR(50),

CONSTRAINT \_activites\_pk

PRIMARY KEY (\_lib-activite),

CONSTRAINT \_activite\_pk1\_competences FOREIGN KEY (numero\_sem)

REFERENCES \_competences(lib\_competence)

);

CREATE TABLE \_ue

(

code\_ue : VARCHAR (50),

numero\_sem : VARCHAR(50),

lib\_activite : VARCHAR(50),

CONSTRAINT \_ue\_pk

PRIMARY KEY (code\_ue),

CONSTRAINT \_ue\_fk1\_semestre FOREIGN KEY (numero\_sem)

REFERENCES \_semestre(numero\_sem),

CONSTRAINT \_ue\_fk2\_activites FOREIGN KEY (lib\_activite)

REFERENCES \_activites(lib\_activite)

);

CREATE TABLE \_niveau

(

numero\_N INTEGER,

numero\_sem VARCHAR(50),

CONSTRAINT \_niveau\_pk

PRIMARY KEY (numero\_N)

);

create TABLE \_correspond

(

code\_P VARCHAR(50),

numero\_N INTEGER,

lib\_activite VARCHAR(50),

CONSTRAINT \_correspond\_pk

PRIMARY KEY (code\_P,numero\_N,lib\_activite)

CONSTRAINT \_correspond\_fk1 FOREIGN KEY (

);