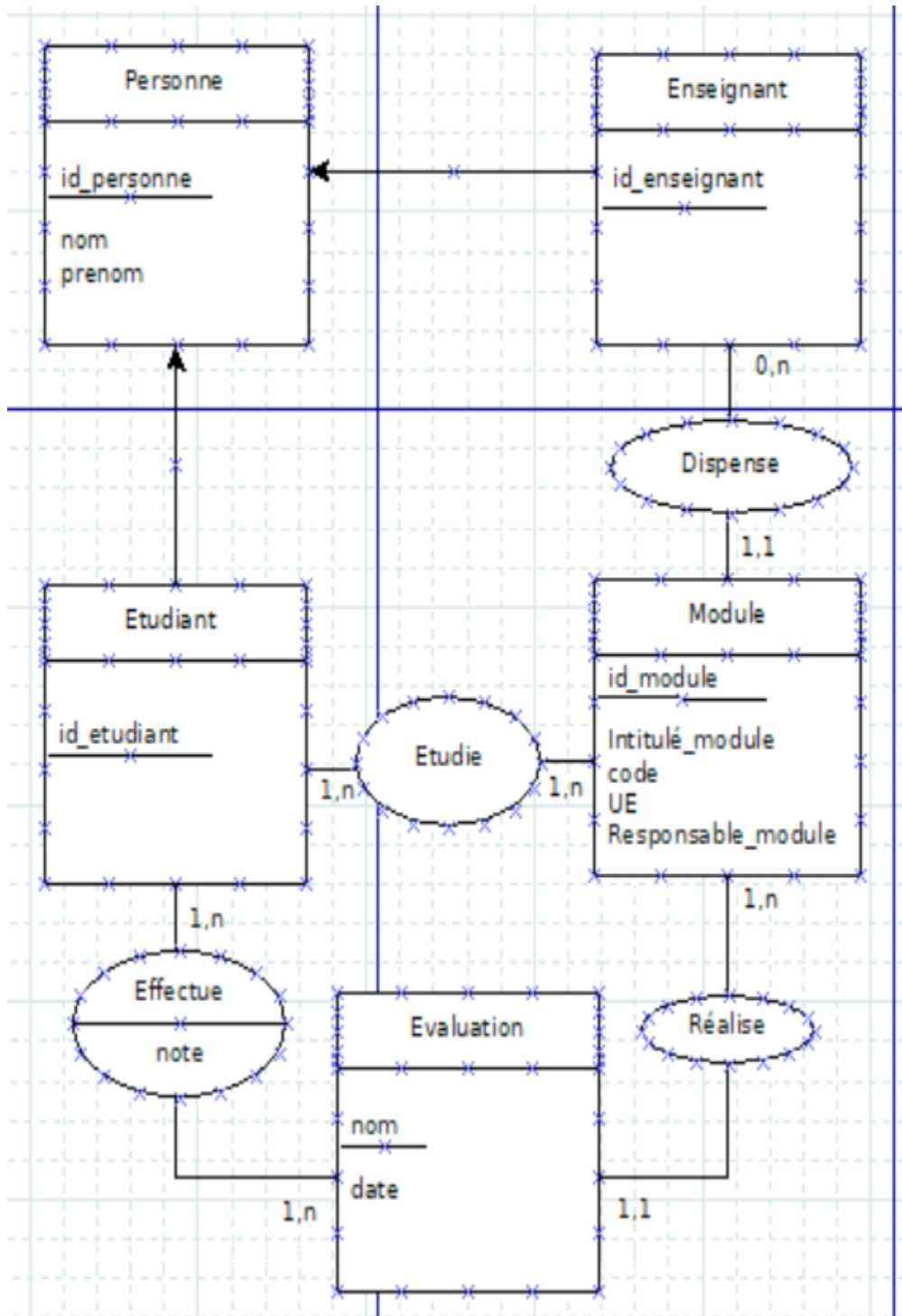


DELLUC
Axel
BUT 1 Info
INDRA
2022/2023

SAE Base de données

I/
1)



2/

Personne(id_personne, nom , prénom)
Enseignant(id_enseignant)
Étudiant(id_etudiant)
Module(id_module , intitulé_module , code , UE , responsable_module)
Evaluation(nom , date)

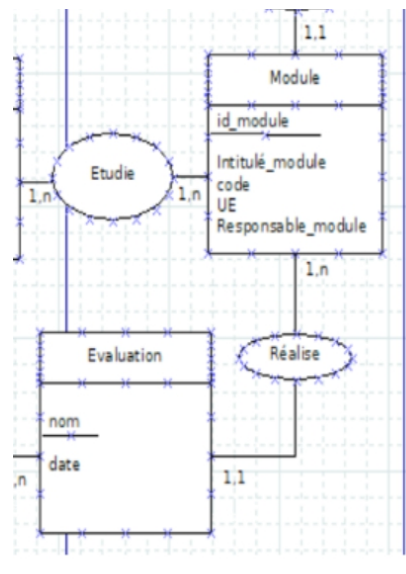
3/

```
CREATE TABLE personne (  
    id_personne INTEGER NOT NULL ,  
    nom VARCHAR NOT NULL,  
    prenom VARCHAR NOT NULL ,  
    PRIMARY KEY (id_personne)  
);  
  
CREATE TABLE enseignant (  
    id_enseignant INTEGER NOT NULL,  
    PRIMARY KEY ( id_enseignant ),  
    FOREIGN KEY ( id_enseignant ) REFERENCES personne(id_personne)  
);  
  
CREATE TABLE etudiant (  
    id_etudiant INTEGER NOT NULL,  
    PRIMARY KEY ( id_etudiant ),  
    FOREIGN KEY ( id_etudiant ) REFERENCES personne(id_personne)  
);  
  
CREATE TABLE module (  
    id_module INTEGER NOT NULL ,  
    intitulé_module VARCHAR NOT NULL,  
    code INTEGER NOT NULL ,  
    UE VARCHAR NOT NULL ,  
    responsable_module INTEGER NOT NULL,  
    PRIMARY KEY ( id_module ),  
    FOREIGN KEY ( responsable_module ) REFERENCES personne(id_personne)  
);  
  
CREATE TABLE evaluation (  
    nom VARCHAR ,  
    date DATE NOT NULL,  
    etudiant INTEGER NOT NULL,  
    note INTEGER NOT NULL,  
    module VARCHAR NOT NULL,  
    PRIMARY KEY ( nom ),  
    FOREIGN KEY (etudiant ) REFERENCES personne(id_personne)  
    FOREIGN KEY ( module) REFERENCES module(id_module)  
);
```

II/

1/

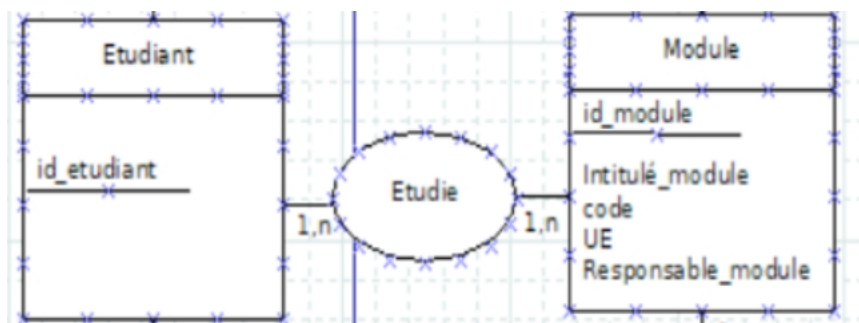
Les associations fonctionnelles , ou associations hiérarchiques sont représentées par un type association liant une cardinalité maximale égale a 1 et une cardinalité maximale égale a n .



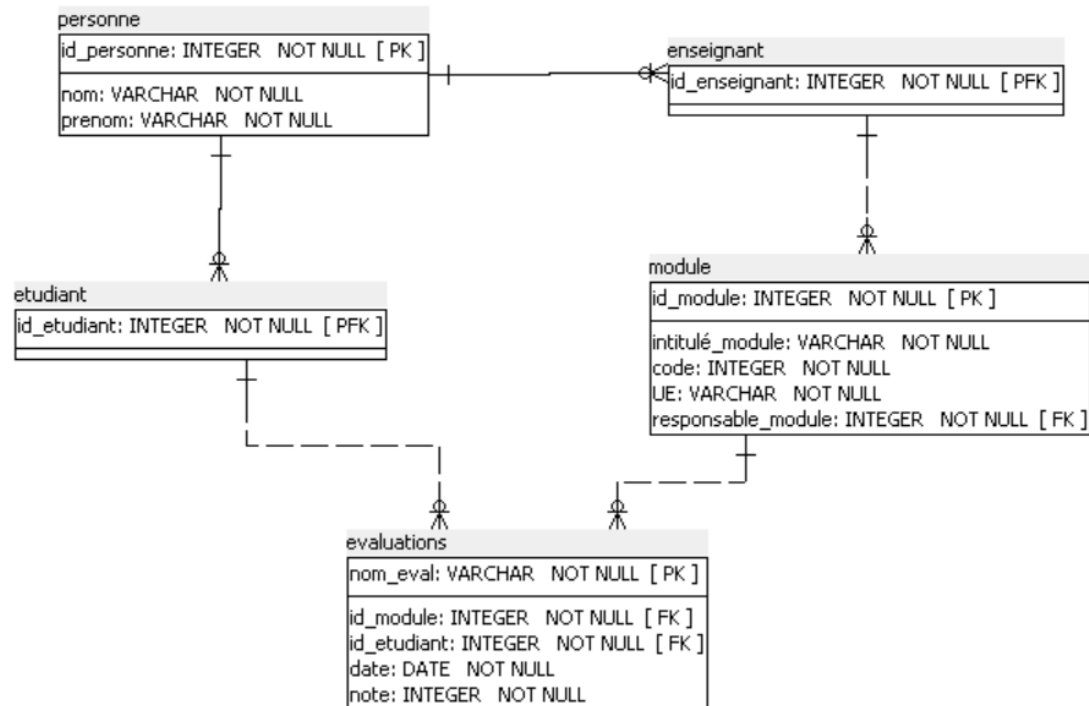
Ici le type-association « Réalise » est une association fonctionnelle puisque la cardinalité est de type [1,n]

2/

Les associations maillées , en majorité présente dans la représentation de ce problème via un modèle entité association , sont des associations liant deux cardinalités maximales égales a n .



Ici le type-association « Etudie » est une association maillée puisque la cardinalité est de type [n,n]



```

CREATE TABLE Personne (
    id_personne INTEGER NOT NULL,
    nom VARCHAR NOT NULL,
    prenom VARCHAR NOT NULL,
    CONSTRAINT id_personne PRIMARY KEY (id_personne)
);
  
```

```

CREATE TABLE etudiant (
    id_etudiant INTEGER NOT NULL,
    CONSTRAINT id_etudiant_ PRIMARY KEY (id_etudiant)
);
  
```

```

CREATE TABLE Enseignant (
    id_enseignant INTEGER NOT NULL,
    CONSTRAINT id_enseignant PRIMARY KEY (id_enseignant)
);
  
```

```

CREATE TABLE module (
    id_module INTEGER NOT NULL,
  
```

```
        intitul_module VARCHAR NOT NULL,  
        code INTEGER NOT NULL,  
        UE VARCHAR NOT NULL,  
        responsable_module INTEGER NOT NULL,  
        CONSTRAINT id_module PRIMARY KEY (id_module)  
    );
```

```
CREATE TABLE evaluations (  
    nom_eval VARCHAR NOT NULL,  
    id_module INTEGER NOT NULL,  
    id_etudiant INTEGER NOT NULL,  
    date DATE NOT NULL,  
    note INTEGER NOT NULL,  
    CONSTRAINT nom PRIMARY KEY (nom_eval)  
);
```

```
ALTER TABLE Enseignant ADD CONSTRAINT personne_enseignant_fk  
FOREIGN KEY (id_enseignant)  
REFERENCES Personne (id_personne)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE etudiant ADD CONSTRAINT personne_etudiant_fk  
FOREIGN KEY (id_etudiant)  
REFERENCES Personne (id_personne)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE evaluations ADD CONSTRAINT etudiant_evaluations_fk  
FOREIGN KEY (id_etudiant)  
REFERENCES etudiant (id_etudiant)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE module ADD CONSTRAINT enseignant_module_fk  
FOREIGN KEY (responsable_module)  
REFERENCES Enseignant (id_enseignant)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE evaluations ADD CONSTRAINT module_evaluations_fk  
FOREIGN KEY (id_module)  
REFERENCES module (id_module)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

5/

La première différence notable est que le script produit automatiquement est beaucoup plus long et contient par conséquent nombres d'informations inutiles . En effet pour les informations complexes que représentent les clés étrangères , le logiciel privilégie de pratiquer ses créations de table en deux temps , la création puis la modification. Augmentant donc par conséquent le nombre de lignes .

III/

```
COPY personne ( id_etudiant , nom_etudiant , prenom_etudiant ) AS ( id_personne , nom , prenom ) FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

```
COPY personne ( id_enseignant , nom_enseignant , prenom_enseignant ) AS ( id_personne , nom , prenom ) FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

```
COPY etudaint ( id_etudiant ) FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

```
COPY enseignant ( id_enseignant )FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

```
COPY module ( id_module , intitule_module , code, ue , id_enseignant ) AS ( id_module , intitule_module , code , ue , responsable_module ) FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

```
COPY evaluation ( nom, date_evaluation , id_module, id_etudiant , note ) FROM 'C:\Users\Utilisateur\Téléchargements\SAE104_data.csv' WITH (FORMAT CSV, HEADER, DELIMITER';');
```

Les deux premières commandes permettent de remplir une table personne regroupant les noms prenom et id des étudiants et des enseignants , les deux suivantes permettent de remplir les tables etudiant et enseignant afin d'avoir la capacité de différencier ceux ci . Enfin les deux dernières requêtes remplissent la table module pour pouvoir identifier celui ci ainsi que le professeur qui en est le responsable . Enfin la table évaluation permet de différencier et de stocker toutes les évaluations et les notes des élèves .

NB: Ces commandes n'ont pas pu être testées a cause du logiciel postgresql qui me refuse l'accès .

```
SELECT nom , prenom , note FROM personne JOIN etudiant ON  
(id_personne=id_etudiant) NATURAL JOIN evaluation USING (id_etudiant ) WHERE  
note<10 ;
```

Ici nous observons une requête permettant de recueillir nom , prenom et note des élèves qui se trouvent avoir raté leurs controles et obtenu des notes strictement en dessous de la moyenne .

```
SELECT nom , prenom,id_personne FROM personne JOIN enseignant ON  
id_personne=id_enseignant JOIN module ON id_enseignant=responsable_module WHERE  
intitule_module='Mathématiques discrètes';
```

Ici nous observons une requête permettant de connaître le nom , prenom et l'id de l'enseignant responsable du module de Mathématiques discrètes .

Le logiciel renvoie ici la table :

Donizeau | Leon | 157