

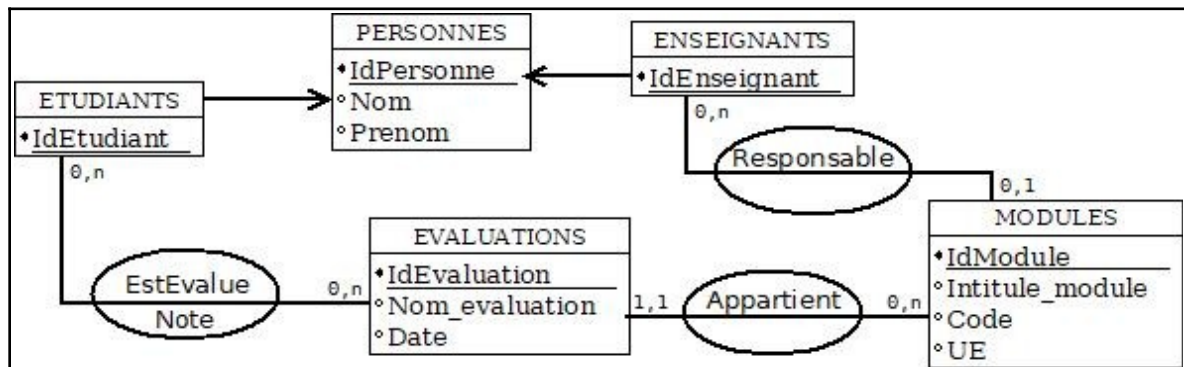
SAÉ S 1.04

Création d'une base de donnée

- 1) Modélisation et script de création « sans AGL »
- 2) Modélisation et script de création « avec AGL »
- 3) Peuplement des tables et requêtes

I / Modélisation et script de création « sans AGL » :

1.1 Modèle Entités-Associations :



1.2 Schéma Relationnel :

PERSONNES(IdPersonne, Nom, Prenom)

ETUDIANTS(IdEtudiant#) référence à IdPersonne de PERSONNE

ENSEIGNANTS(IdEnseignant#) référence à IdPersonne de PERSONNE

MODULES(IdModule, Intitule_module, Code, UE, IdEnseignant#) référence à ENSEIGNANTS

EVALUATIONS(IdEvaluation, Nom_evaluation, Date, IdModule#) référence à MODULES

ESTEVALUE(IdEtudiant#, IdEvaluation#, Note) référence à ETUDIANTS et à EVALUATIONS

1.3 Script SQL :

```
CREATE TABLE personnes (  
    IdPersonne INTEGER PRIMARY KEY,  
    Nom VARCHAR NOT NULL,  
    Prenom VARCHAR NOT NULL);
```

```
CREATE TABLE etudiants (  
    IdEtudiant INTEGER PRIMARY KEY REFERENCES personnes (IdPersonne) );
```

```
CREATE TABLE enseignants (  
    IdEnseignant INTEGER PRIMARY KEY REFERENCES personnes (IdPersonne) );
```

```
CREATE TABLE modules (  
    IdModule INTEGER PRIMARY KEY,  
    Intitule_module VARCHAR NOT NULL,  
    Code VARCHAR NOT NULL,  
    UE VARCHAR NOT NULL,  
    IdEnseignant INTEGER REFERENCES enseignants (IdEnseignant) );
```

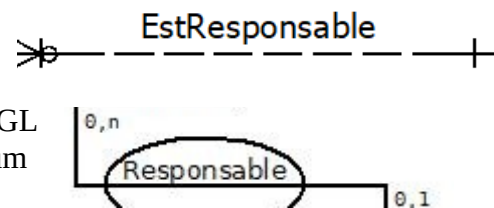
```
CREATE TABLE evaluations (
    IdEvaluation INTEGER PRIMARY KEY,
    Nom_Evaluation VARCHAR NOT NULL,
    Date DATE NOT NULL,
    IdModule INTEGER REFERENCES modules (IdModule) );
```

```
CREATE TABLE estevalue (
    IdEtudiant INTEGER REFERENCES etudiants (IdEtudiant),
    IdEvaluation INTEGER REFERENCES evaluations (IdEvaluation),
    Note FLOAT CHECK (note > 0 OR note IS NULL),
    PRIMARY KEY (IdEtudiant, IdEvaluation) );
```

II / Modélisation et script de création « avec AGL » :

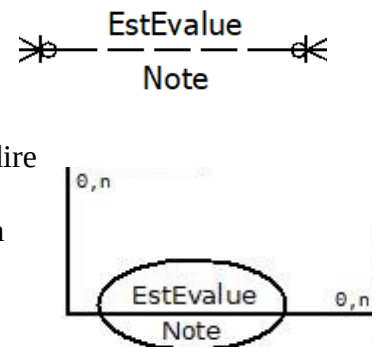
2.1 Illustrations comparatives association fonctionnelle :

L'association fonctionnelle est représenté de manière différente sur l'AGL et à la main. Une association fonctionnelle à une cardinalité de maximum n allant vers maximum 1. De ce fait, à la main on note (0,n) et (0,1) et sur l'AGL (0,n) est représenté par une flèche, un trait et un rond et (0,1) par un trait et un rond. EstResponsable est une association fonctionnelle.

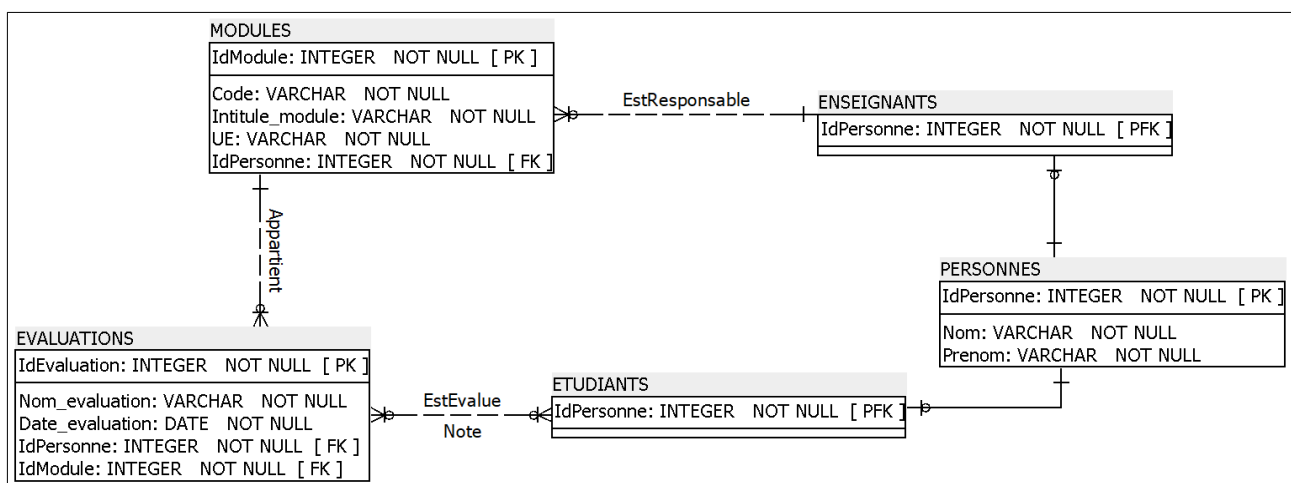


2.2 Illustrations comparatives association maillée :

L'association maillée est de la même manière que la fonctionnelle c'est-à-dire qu'elles sont représenté différemment, sa cardinalité est de (0,n) de chaque côté. A la main, ils sont représenté par (0,n) et sur l'AGL par une flèche, un trait et un rond. EstEvalue en est une.



2.3 Modèle Entités-Associations :



2.4 Script SQL généré :

```
CREATE TABLE public.PERSONNES (  
    IdPersonne INTEGER NOT NULL,  
    Nom VARCHAR NOT NULL,  
    Prenom VARCHAR NOT NULL,  
    CONSTRAINT personnes_pk PRIMARY KEY (IdPersonne)  
);
```

```
CREATE TABLE public.ETUDIANTS (  
    IdPersonne INTEGER NOT NULL,  
    CONSTRAINT etudiants_pk PRIMARY KEY (IdPersonne)  
);
```

```
CREATE TABLE public.ENSEIGNANTS (  
    IdPersonne INTEGER NOT NULL,  
    CONSTRAINT enseignants_pk PRIMARY KEY (IdPersonne)  
);
```

```
CREATE TABLE public.MODULES (  
    IdModule INTEGER NOT NULL,  
    Code VARCHAR NOT NULL,  
    Intitule_module VARCHAR NOT NULL,  
    UE VARCHAR NOT NULL,  
    IdPersonne INTEGER NOT NULL,  
    CONSTRAINT modules_pk PRIMARY KEY (IdModule)  
);
```

```
CREATE TABLE public.EVALUATIONS (  
    IdEvaluation INTEGER NOT NULL,  
    Nom_evaluation VARCHAR NOT NULL,  
    Date_evaluation DATE NOT NULL,  
    IdPersonne INTEGER NOT NULL,  
    IdModule INTEGER NOT NULL,  
    CONSTRAINT evaluations_pk PRIMARY KEY (IdEvaluation)  
);
```

```
ALTER TABLE public.ETUDIANTS ADD CONSTRAINT personne_etudiant_fk  
FOREIGN KEY (IdPersonne)  
REFERENCES public.PERSONNES (IdPersonne)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.ENSEIGNANTS ADD CONSTRAINT personne_enseignant_fk  
FOREIGN KEY (IdPersonne)  
REFERENCES public.PERSONNES (IdPersonne)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION
```

NOT DEFERRABLE;

```
ALTER TABLE public.EVALUATIONS ADD CONSTRAINT etudiant_evaluation_fk
FOREIGN KEY (IdPersonne)
REFERENCES public.ETUDIANTS (IdPersonne)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE public.MODULES ADD CONSTRAINT enseignant_module_fk
FOREIGN KEY (IdPersonne)
REFERENCES public.ENSEIGNANTS (IdPersonne)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE public.EVALUATIONS ADD CONSTRAINT module_evaluation_fk
FOREIGN KEY (IdModule)
REFERENCES public.MODULES (IdModule)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

2.5 Différences entre les scripts :

Le script à la main est plus court puisqu'il est plus condensé, par exemple, pour indiquer une clé primaire il nous suffit d'ajouter PRIMARY KEY (IdPersonne INTEGER PRIMARY KEY) alors que le script généré ajoute une autre ligne de contrainte (CONSTRAINT personnes_pk PRIMARY KEY (IdPersonne)). Mais ce n'est pas la seule chose, pour une clé étrangère il nous faut simplement mettre REFERENCES contrairement au script qui ajoute un bloc entier de six lignes. De plus, lorsque nous écrivons un script à la main il nous est possible d'oublier certaines choses ou de ne pas respecter la contrainte de domaine en se trompant de type, de même pour la contrainte d'intégrité de référence. Si nous créons pas les tables dans le bon ordre il se peut que le script soit erroné suite à l'inversement de la création de certaines tables or celui généré automatiquement n'aura aucun problème de contrainte. Pour finir, celui généré est plus simple et plus rapide étant donné que si notre Entité-Association possède des erreurs, il nous les indique et nous propose de les corriger s'il lui est possible.

III / Peuplement des tables et requêtes :

3.1 Différentes étapes de peuplement :

Dans un premier temps, j'ai créé un script après avoir créé une nouvelle base, puis j'y ai copié mon script créant les différentes tables fait manuellement. A la fin de ce dernier, j'ai créé une nouvelle table (temporaire) avec pour nom de colonnes celles utilisées dans le fichier data fourni pour cette saé.

```
CREATE TABLE temporaire (
    id_enseignant INTEGER,
    nom_enseignant VARCHAR,
    prenom_enseignant VARCHAR,
    id_module INTEGER,
    code VARCHAR,
    ue VARCHAR,
```

```
intitule_module VARCHAR,  
id_evaluation INTEGER,  
nom_evaluation VARCHAR,  
date_evaluation DATE,  
note FLOAT,  
id_etudiant INTEGER,  
nom_etudiant VARCHAR,  
prenom_etudiant VARCHAR );
```

J'ai par la suite copié les données du fichier data.csv dans la table temporaire grâce à la commande COPY.

```
COPY temporaire FROM 'chemin data.csv' DELIMITER ';' CSV Header;
```

Une fois avoir rempli cette table, il suffit de compléter les premières tables créé à l'aide de la commande INSERT INTO SELECT DISTINCT et FROM

```
INSERT INTO personnes (SELECT DISTINCT id_enseignant,nom_enseignant,prenom_enseignant FROM  
temporaire);  
INSERT INTO personnes (SELECT DISTINCT id_etudiant,nom_etudiant,prenom_etudiant FROM temporaire);  
INSERT INTO enseignants (SELECT DISTINCT id_enseignant FROM temporaire);  
INSERT INTO etudiants (SELECT DISTINCT id_etudiant FROM temporaire);  
INSERT INTO modules (SELECT DISTINCT id_module,intitule_module,ue,code, id_enseignant FROM temporaire);  
INSERT INTO evaluations (SELECT DISTINCT id_evaluation,nom_evaluation,date_evaluation,id_module FROM  
temporaire);  
INSERT INTO estevalue (SELECT DISTINCT id_etudiant, id_evaluation, note FROM temporaire);
```

3.2 Deux requêtes intéressantes :

```
SELECT DISTINCT personnes.prenom FROM etudiants, personnes, estevalue WHERE etudiants.IdEtudiant  
= personnes.IdPersonne AND etudiants.IdEtudiant = estevalue.IdEtudiant AND estevalue.Note >=10 AND  
estevalue.Note <= 15;
```

Cette commande permet de sélectionner les différents prénoms des étudiants ayant eu une note entre 10 et 15 compris

```
SELECT DISTINCT modules.intitule_module FROM modules, evaluations WHERE modules.IdModule =  
evaluations.IdModule AND evaluations.date = '2021-09-06';
```

Celle-ci permet de permettre d'afficher le nom des modules ayant fait une évaluation le 06 septembre 2021.