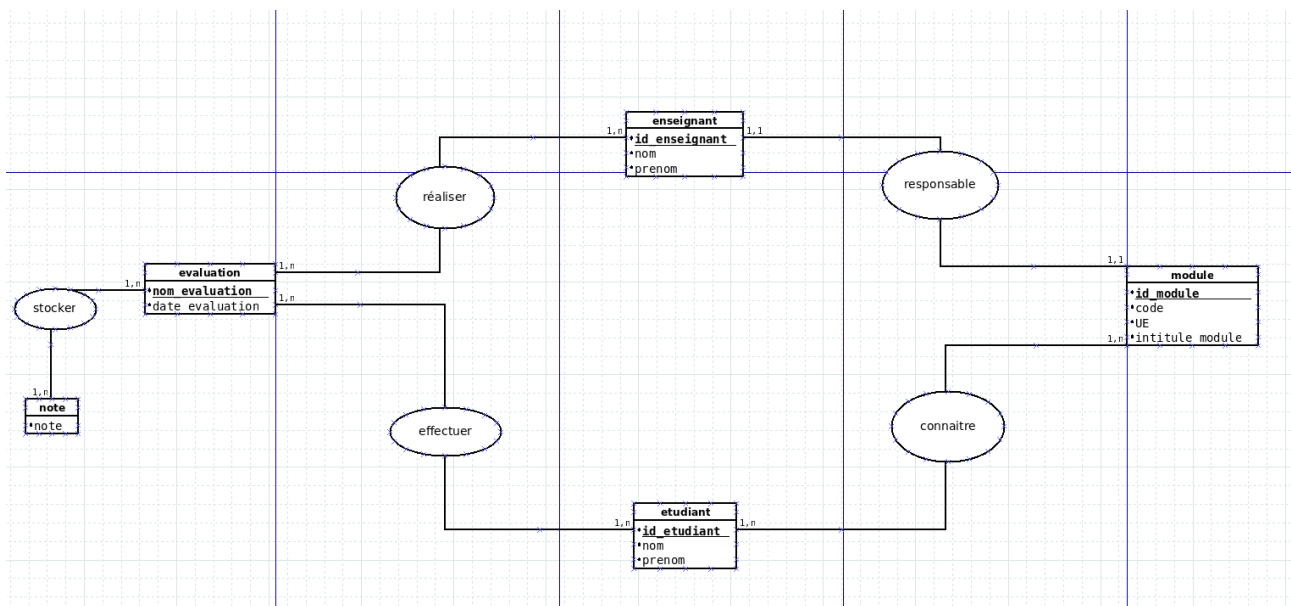


SAE Bases de Données et langage SQL
Création d'une base de données : notations.

2.1) Modélisation et script de création «sans AGL »

1)



2)

Enseignant(id_enseignant,nom_etudiant,prenom_etudiant)
 Étudiant(id_etudiant,nom_etudiant,prenom_etudiant)
 Module(id_module,code,UE,intitule_module, #id_enseignant)
 Évaluation(nom_evaluation,date_evaluation)
 Note(nom_evaluation,id_etudiant,id_module,note)

3)

```

CREATE TABLE Enseignant (
    id_enseignant INTEGER PRIMARY KEY,
    nom_enseignant VARCHAR NOT NULL,
    prenom_enseignant VARCHAR NOT NULL

);
    
```

```

CREATE TABLE etudiant (
    id_etudiant INTEGER PRIMARY KEY,
    nom_etudiant VARCHAR NOT NULL,
    prenom_etudiant VARCHAR NOT NULL

);

CREATE TABLE Module (
    id_module INTEGER PRIMARY KEY,
    id_enseignant INTEGER REFERENCES enseignant,
    intitule_module VARCHAR NOT NULL,
    code VARCHAR NOT NULL,
    UE VARCHAR NOT NULL

);

CREATE TABLE Evaluation (
    nom_evaluation VARCHAR PRIMARY KEY,
    id_enseignant INTEGER REFERENCES enseignant,
    date_evaluation DATE NOT NULL,

);

CREATE TABLE Note (
    nom_evaluation VARCHAR PRIMARY KEY,
    id_etudiant INTEGER NOT NULL,
    id_module INTEGER NOT NULL,
    note FLOAT
);

```

2,2) Modélisation et script de création «avec AGL».

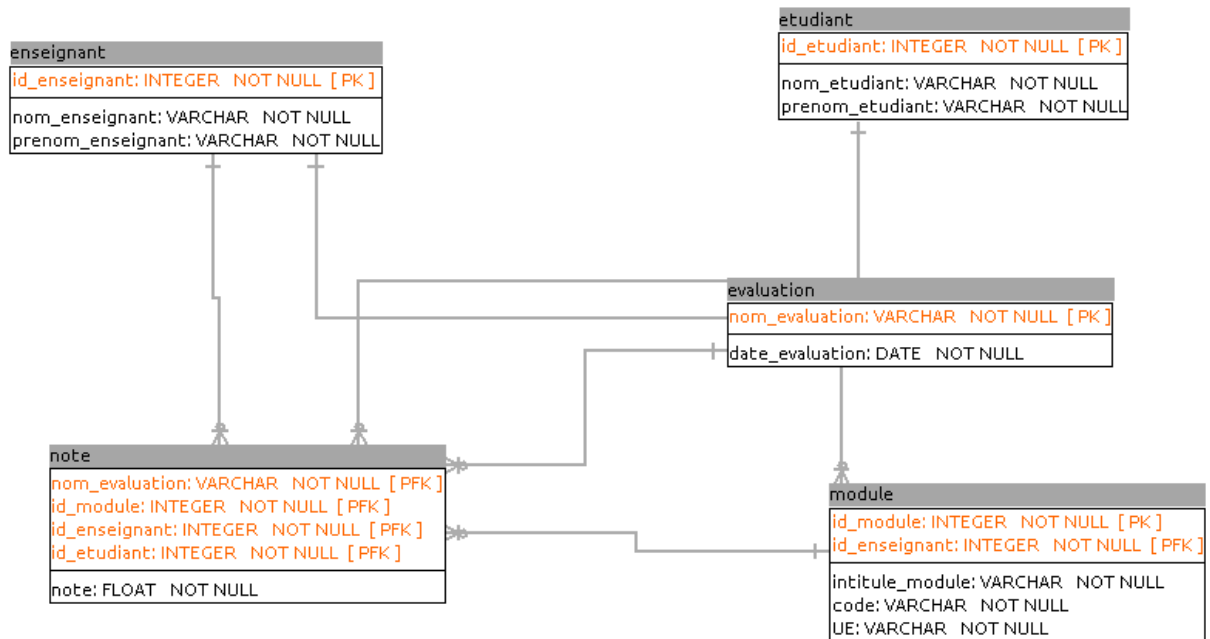
1)

Un type association fonctionnelle aura une cardinalité de 1,n.

2)

Un type association maillé aura une cardinalité de n,n.

3)



4)

```

CREATE TABLE evaluation (
    nom_evaluation VARCHAR NOT NULL,
    date_evaluation DATE NOT NULL,
    CONSTRAINT nom_evaluation PRIMARY KEY (nom_evaluation)
);
  
```

```

CREATE SEQUENCE module_id_module_seq;
  
```

```

CREATE TABLE module (
    id_module INTEGER NOT NULL DEFAULT nextval('module_id_module_seq'),
    intitule_module VARCHAR NOT NULL,
    code VARCHAR NOT NULL,
    UE VARCHAR NOT NULL,
    CONSTRAINT id_module PRIMARY KEY (id_module)
);
  
```

```

ALTER SEQUENCE module_id_module_seq OWNED BY module.id_module;
  
```

```

CREATE SEQUENCE etudiant_id_etudiant_seq;
  
```

```
CREATE TABLE etudiant (  
    id_etudiant INTEGER NOT NULL DEFAULT nextval('etudiant_id_etudiant_seq'),  
    nom_etudiant VARCHAR NOT NULL,  
    prenom_etudiant VARCHAR NOT NULL,  
    CONSTRAINT id_etudiant PRIMARY KEY (id_etudiant)  
);
```

```
ALTER SEQUENCE etudiant_id_etudiant_seq OWNED BY etudiant.id_etudiant;
```

```
CREATE SEQUENCE enseignant_id_enseignant_seq;
```

```
CREATE TABLE enseignant (  
    id_enseignant INTEGER NOT NULL DEFAULT  
nextval('enseignant_id_enseignant_seq'),  
    nom_enseignant VARCHAR NOT NULL,  
    prenom_enseignant VARCHAR NOT NULL,  
    CONSTRAINT id_enseignant PRIMARY KEY (id_enseignant)  
);
```

```
ALTER SEQUENCE enseignant_id_enseignant_seq OWNED BY enseignant.id_enseignant;
```

```
CREATE TABLE note (  
    nom_evaluation VARCHAR NOT NULL,  
    id_module INTEGER NOT NULL,  
    id_enseignant INTEGER NOT NULL,  
    id_etudiant INTEGER NOT NULL,  
    note REAL NOT NULL,  
    CONSTRAINT nom_evaluation PRIMARY KEY (nom_evaluation, id_module,  
id_enseignant, id_etudiant)  
);
```

```
ALTER TABLE note ADD CONSTRAINT evaluation_note_fk  
FOREIGN KEY (nom_evaluation)  
REFERENCES evaluation (nom_evaluation)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

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

```
ALTER TABLE note ADD CONSTRAINT etudiant_note_fk  
FOREIGN KEY (id_etudiant)  
REFERENCES etudiant (id_etudiant)  
ON DELETE NO ACTION
```

ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE note ADD CONSTRAINT enseignant_note_fk
FOREIGN KEY (id_enseignant)
REFERENCES enseignant (id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

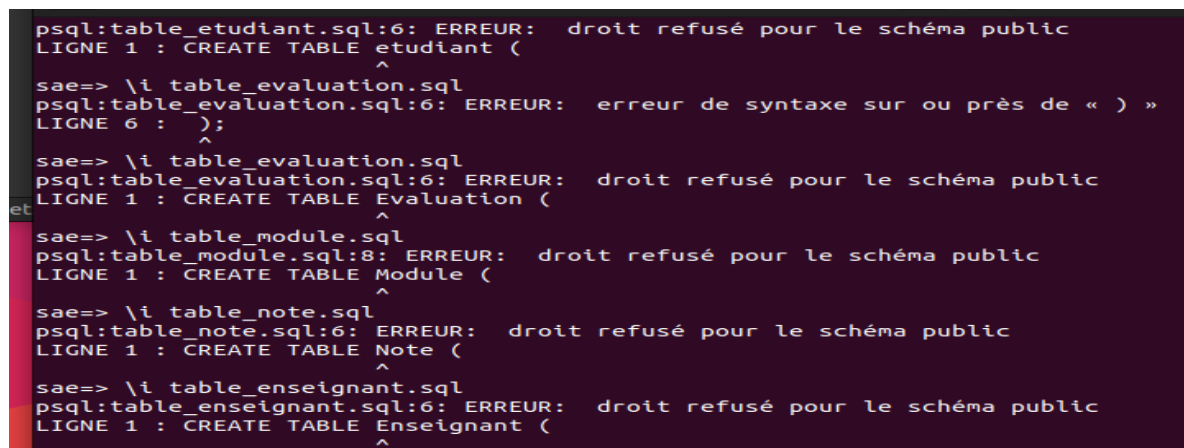
5)

On ne retrouve pas de si grande différence entre le script tapé manuellement et le script fournis automatiquement néanmoins, AGL utilisent des contraintes. Et l'AGL se permet de rajouter certaines autres fonctionnalité que si on le faisait manuellement comme ALTER TABLE.

2,3) Peuplement des tables et requêtes.

1)

Malheureusement je n'arrive pas à créer mes tables donc impossible de pouvoir copier les données situé dans le fichier data.csv. Pourtant le script pour créer les tables est bon mais je ne sais pas pourquoi cela ne fonctionne pas.



```
psql:table_etudiant.sql:6: ERREUR:  droit refusé pour le schéma public
LIGNE 1 : CREATE TABLE etudiant (
      ^
sae=> \i table_evaluation.sql
psql:table_evaluation.sql:6: ERREUR:  erreur de syntaxe sur ou près de « ) »
LIGNE 6 : );
      ^
sae=> \i table_evaluation.sql
psql:table_evaluation.sql:6: ERREUR:  droit refusé pour le schéma public
LIGNE 1 : CREATE TABLE Evaluation (
      ^
sae=> \i table_module.sql
psql:table_module.sql:8: ERREUR:  droit refusé pour le schéma public
LIGNE 1 : CREATE TABLE Module (
      ^
sae=> \i table_note.sql
psql:table_note.sql:6: ERREUR:  droit refusé pour le schéma public
LIGNE 1 : CREATE TABLE Note (
      ^
sae=> \i table_enseignant.sql
psql:table_enseignant.sql:6: ERREUR:  droit refusé pour le schéma public
LIGNE 1 : CREATE TABLE Enseignant (
      ^
```

2)

SELECT nom,prenom,intitule_module FROM enseignant,module WHERE
module.id_module=module.enseignant ;

Cette requête permet de connaître le nom et prénom de l'enseignant et l'intitule du module qu'il enseigne.

```
SELECT note,nom_evaluation FROM note,evaluation ;
```

Permet de connaître les notes et le nom des évaluations effectuer.