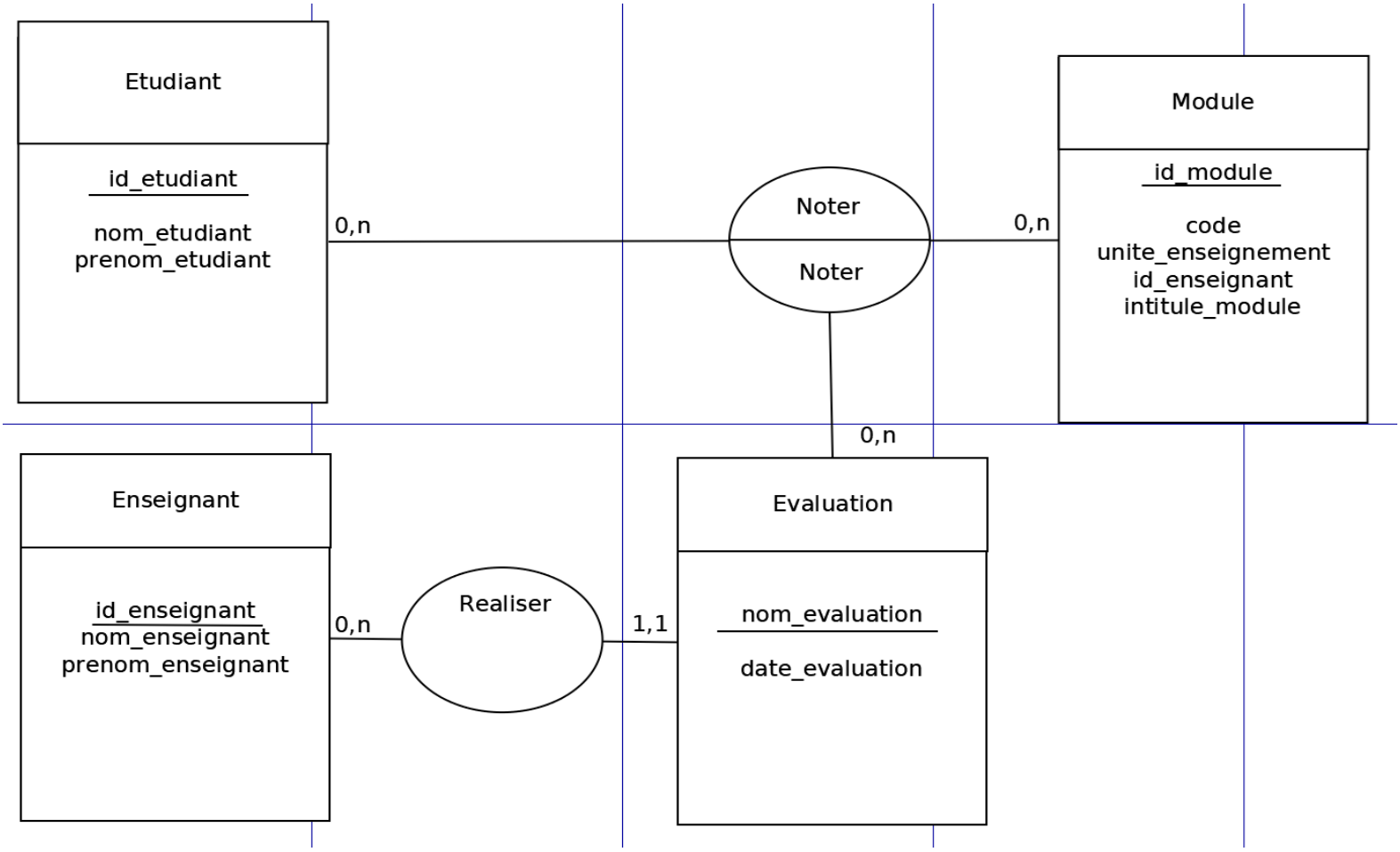
Mathis
Verchere
BUT INFO 1
INDRA
2022/2023

SAE Bases de données et langage sql.

Creation d'un base de données notations.

## Modélisation et script de création «  sans AGL ».

### 1)




### 2)

Etudiant (id_etudiant, nom_etudiant, prenom etudiant)
Module (id_module, #id_enseignant, intitule_module, code, unite_enseignement)
Enseignant (id_enseignant, nom_enseignant, prenom_enseignant)
Evaluation (nom_evalution, date_evaluation)
Note (nom_evaluation ,id_module, nom_evaluation ,id_etudiant,note)

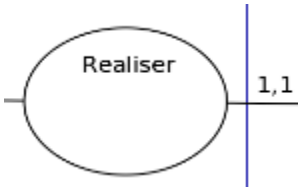
**3)**

```
CEATE TABLE Enseignant (
        id_enseignant INTEGER PRIMARY KEY,
        nom_enseignant VARCHAR NOT NULL,
        prenom_enseignant VARCHAR NOT NULL
) ;

CEATE TABLE Module (
        id_module INTEGER PRIMARY KEY,
        id_enseignant INTEGER REFERENCES Enseignant ON DELETE SET NULL,
        intitule_module VARCHAR NOT NULL,
        code VARCHAR NOT NULL,
        unite_enseignement VARCHAR NOT NULL
) ;

CEATE TABLE Etudiant (
        id_etudiant INTEGER PRIMARY KEY,
        nom_etudiant VARCHAR NOT NULL,
        prenom_etudiant VARCHAR NOT NULL
) ;

CEATE TABLE Evaluation (
        nom_evaluation VARCHAR PRIMARY KEY,
        id_enseignant INTEGER REFERENCES Enseignant ON DELETE SET NULL,
        date_evaluation DATE NOTE NULL
) ;

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

## Modélisation et script de creation « avec AGL ».

**1)**

Association fonctionnel, ajoute simplement une clé étrangère quand la cardinalité maximal est de 1, donc 1,1.

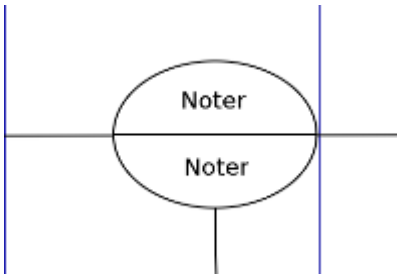



1,1

**2)**

Type association maillée sont des jointures de deux tables avec plusieurs clés etrangères,
Et de cardinalité 0,n.

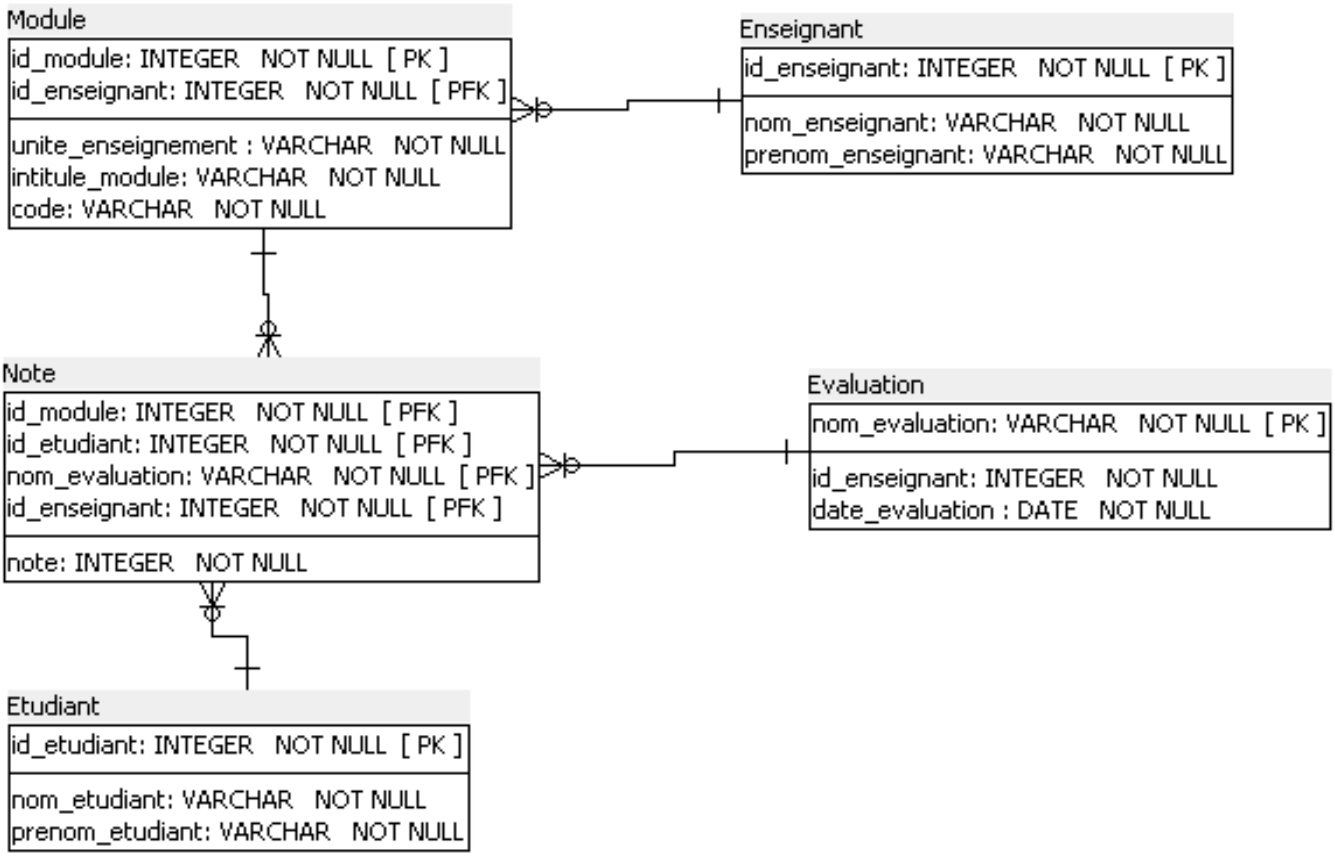
Comme par exemple :




l'AGL n'affiche pas les associations maillése mais elles sont bien presente.

Table(idtable1, idtable2, attributs)

**3)**




**4)**

```
CREATE TABLE Evaluation (
        nom_evaluation VARCHAR NOT NULL,
        id_enseignant INTEGER NOT NULL,
        date_evaluation_ DATE NOT NULL,
        CONSTRAINT nom_evaluation PRIMARY KEY (nom_evaluation)
);
COMMENT ON TABLE Evaluation IS 'evaluation';


CREATE TABLE Etudiant (
        id_etudiant INTEGER NOT NULL,
        nom_etudiant VARCHAR NOT NULL,
        prenom_etudiant VARCHAR NOT NULL,
        CONSTRAINT id_etudiant PRIMARY KEY (id_etudiant)
);
COMMENT ON TABLE Etudiant IS 'etudiant';


CREATE TABLE Enseignant (
```

```
	id_enseignant INTEGER NOT NULL,
	nom_enseignant VARCHAR NOT NULL,
	prenom_enseignant VARCHAR NOT NULL,
	CONSTRAINT id_enseignant PRIMARY KEY (id_enseignant)
);
COMMENT ON TABLE Enseignant IS 'enseignant';


CREATE TABLE Module (
	id_module INTEGER NOT NULL,
	id_enseignant INTEGER NOT NULL,
	unite_enseignement_ VARCHAR NOT NULL,
	intitule_module VARCHAR NOT NULL,
	code VARCHAR NOT NULL,
	CONSTRAINT id_module PRIMARY KEY (id_module, id_enseignant)
);
COMMENT ON TABLE Module IS 'module';


CREATE TABLE Note (
	id_module INTEGER NOT NULL,
	id_etudiant INTEGER NOT NULL,
	nom_evaluation VARCHAR NOT NULL,
	id_enseignant INTEGER NOT NULL,
	note INTEGER NOT NULL,
	CONSTRAINT id_note PRIMARY KEY (id_module, id_etudiant, nom_evaluation,
id_enseignant)
);
COMMENT ON TABLE Note IS 'note';


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 Etudiant_Note_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 (id_enseignant)
REFERENCES Enseignant (id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

ALTER TABLE Note ADD CONSTRAINT Module_Note_fk
FOREIGN KEY (id_module, id_enseignant)
REFERENCES Module (id_module, id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

**5)**

Le script fait manuellement est long a faire mais si on le fait correctement il a la meme forme et meme syntaxe que celui fait automatiquement, le faire automatiquement par AGL est plus rapide, et on a moin de risque de faire des erreurs.
Mais le mieux reste manuellement, lorsqu'on ne sait pas utiliser le AGL.
Et l'AGL a ajouter des ALTER TABLE a la fin.

## Peuplement des tables et requêtes.

**1)**

COPY ENSEIGNANT (id_enseignant, prenom_enseignant, nom_enseignant) FROM 'C:\Users\Utilisateur\Downloads\data.csv' WITH (FORMAT CSV,HEADER,DELIMITER';');

La commande COPY, copie dans la table ENSEIGNANT, les : id_enseignant, prenom_enseignant, nom_enseignant.
Depuis le fichier cible, grace a la commande FROM. Qui ouvre le fichier en format CSV, et delimite les valeurs avec ' ;' grave au WITH().

COPY ETUDIANT (id_etudiant, prenom_etudiant, nom_etudiant) FROM 'C:\Users\Utilisateur\Downloads\data.csv' WITH (FORMAT CSV,HEADER,DELIMITER';');

La commande COPY, copie dans la table ETUDIANT, les : id_etudiant, prenom_etudiant, nom_etudiant.
Depuis le fichier cible, grace a la commande FROM. Qui ouvre le fichier en format CSV, et delimite les valeurs avec ' ;' grave au WITH().

COPY MODULE (id_module, id_enseignant, intitule_module, code, unite_enseignement) FROM 'C:\Users\Utilisateur\Downloads\data.csv' WITH (FORMAT CSV,HEADER,DELIMITER';');

La commande COPY, copie dans la table MODULE, les id_module, id_enseignant, intitule_module, code, unite_enseignement.
Depuis le fichier cible, grace a la commande FROM. Qui ouvre le fichier en format CSV, et delimite les valeurs avec ' ;' grave au WITH().

```
COPY EVALUATION (nom_evaluation, date_evaluation, id_enseignant) FROM 'C:\Users\Utilisateur\Downloads\data.csv' WITH (FORMAT CSV,HEADER,DELIMITER';');
```

La commande COPY, copie dans la table EVALUATION , les :  nom_evaluation, date_evaluation, id_enseignant.
Depuis le fichier cible, grace a la commande FROM. Qui ouvre le fichier en format CSV, et delimite les valeurs avec ' ;' grave au WITH().

```
COPY NOTER (id_module, id_etudiant, nom_evaluation, note) FROM 'C:\Users\Utilisateur\Downloads\data.csv' WITH (FORMAT CSV,HEADER,DELIMITER';');
```

La commande COPY, copie dans la table NOTER, les :  id_module, id_etudiant, nom_evaluation, note.
Depuis le fichier cible, grace a la commande FROM. Qui ouvre le fichier en format CSV, et delimite les valeurs avec ' ;' grave au WITH().

**2)**

```
SELECT nom,prenom FROM enseignant NATURAL JOIN module USING (id_enseignant) ;
```

La requête affiche les noms et prenoms des enseignants qui font partie d'un module, grace au NATURAL JOIN qui est accompagner de USING (id_enseignant) pour bien lié les deux tables.

```
SELECT nom,prenom,note FROM etudiant NATURAL JOIN noter USING (id_etudiant) WHERE note >= 10 ;
```

la requête affiche les noms, prenoms et notes, de tout les élèves qui ont plus de 10, grave a NATURAL JOIN qui utilise le USING(id_etudiant) lie les tables etudiant et noter. Fini de WHERE comme condition pour recuperer les notes superieur ou egalent à 10.