RAPPORT

SAE création d’une base de données

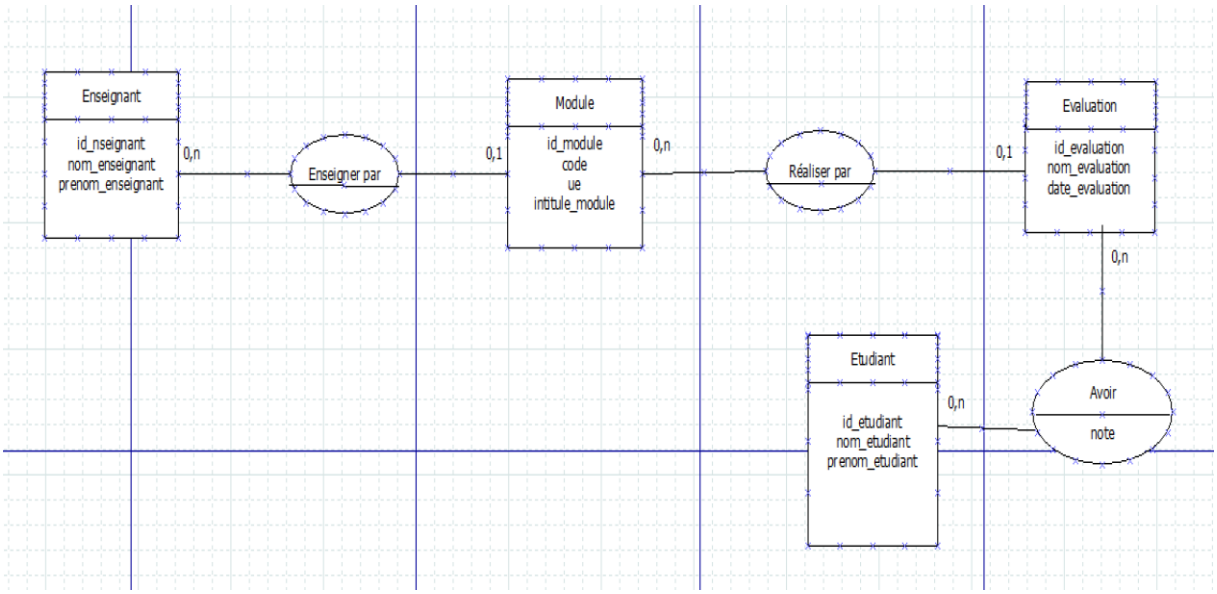
Réalisé par :

Nour elhouda BOUAMLAT en BUT1 informatique, groupe SHANGO

SOMMAIRE :

I. Modélisation et script de création « sans AGL »

1. Le modèle entités-associations de la base de données :



2. Le schéma relationnel :

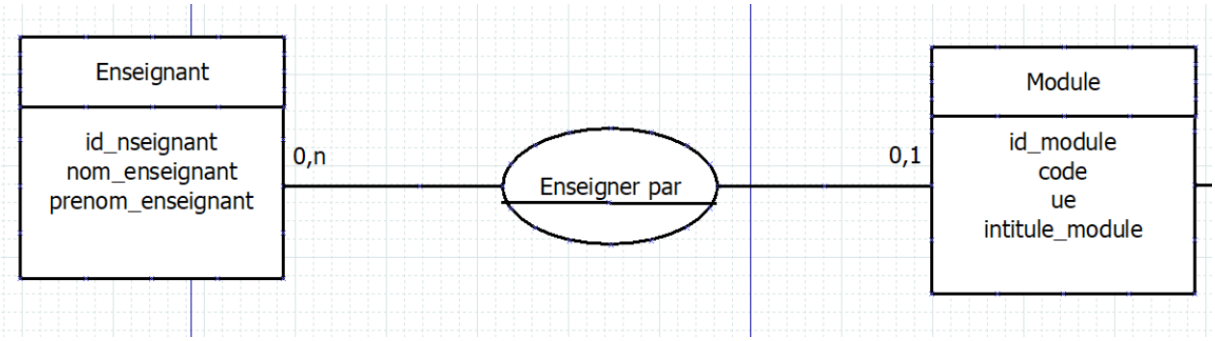
Module (id_module, code, ue, intitule_module, id_enseignant) où id_enseignant fait référence à Enseignant
Enseignant (id_enseignant, nom_enseignant, prenom_enseignant)
Evaluation (id_evaluation, nom_evaluation, date_evaluation, id_module) où id_module fait référence à Module
Etudiant (id_etudiant, nom_etudiant, prenom_etudiant)
Avoir (id_etudiant, id_evaluation, note)

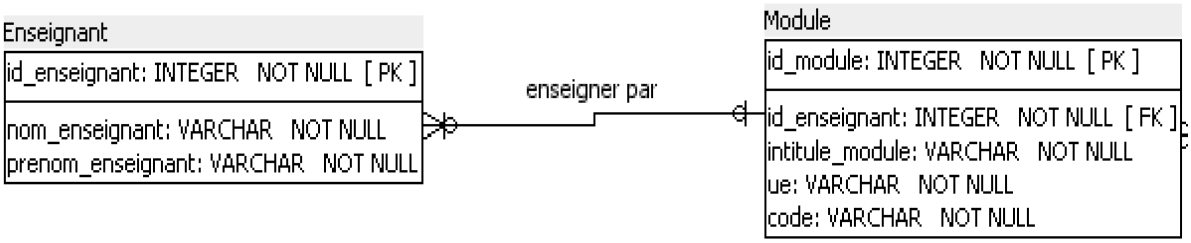
3. Le script SQL de création des tables :

```
CREATE TABLE Enseignant (
        id_enseignant INTEGER PRIMARY KEY,
        nom_enseignant VARCHAR,
        prenom_enseignant VARCHAR
) ;
CREATE TABLE Module (
        id_module INTEGER PRIMARY KEY,
        code VARCHAR,
        ue VARCHAR,
        intitule_module VARCHAR,
        id_enseignant INTEGER REFERENCES Enseignant
) ;
CREATE TABLE Evaluation (
        id_evaluation INTEGER PRIMARY KEY,
        nom_evaluation VARCHAR,
        date_evaluation VARCHAR,
         id_module INTEGER REFERENCES Module
) ;
CREATE TABLE Etudiant (
        id_etudiant INTEGER PRIMARY KEY,
        nom_etudiant VARCHAR,
        prenom_etudiant VARCHAR
) ;
CREATE TABLE Avoir (
        id_etudiant INTEGER REFERENCES Etudiant,
        id_evaluation INTEGER REFERENCES Evaluation,
        PRIMARY KEY (id_etudiant, id_evaluation),
        note VARCHAR
) ;
```

II. Modélisation et script de création « avec AGL »

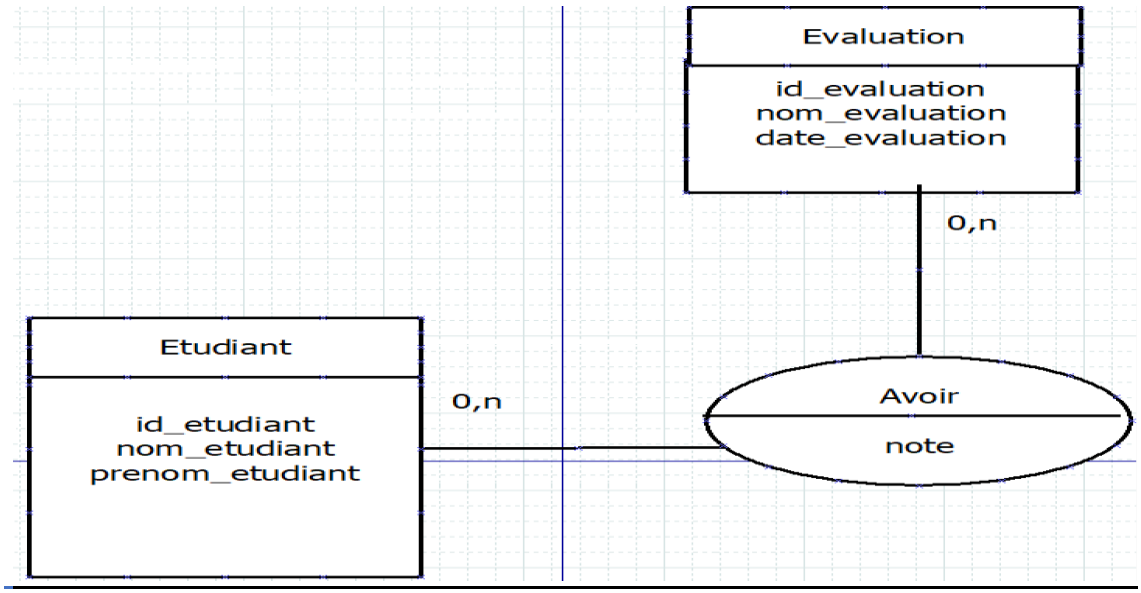
1. Association fonctionnelle cours/AGL :

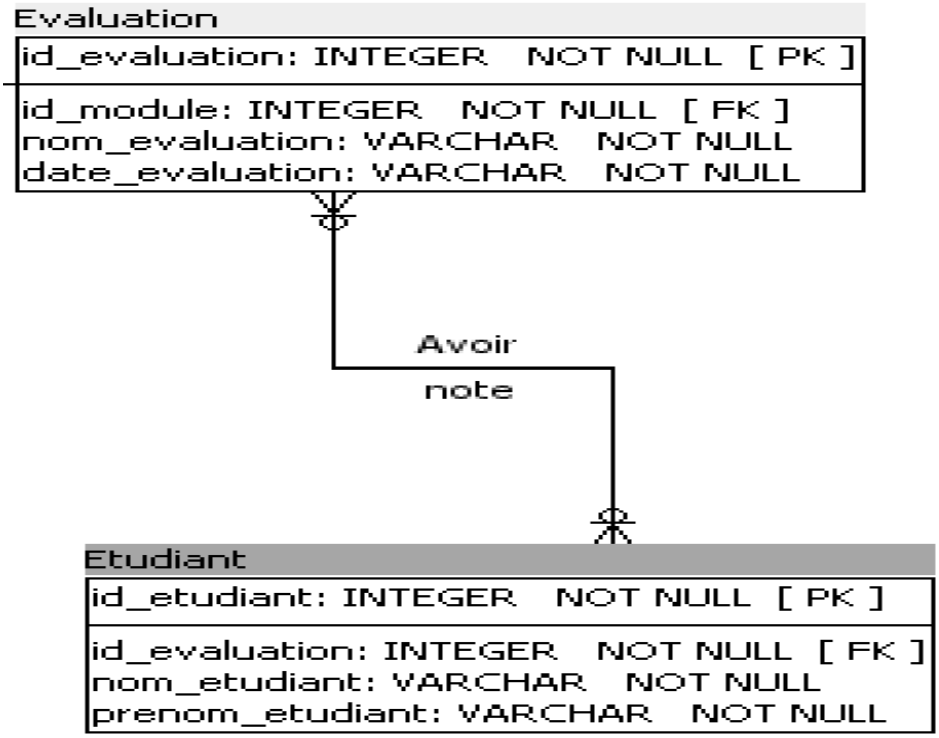




⇨ On a deux associations fonctionnelles, la première faite manuellement sur DIA, et la deuxième avec l'AGL. On remarque l'absence des cardinalités dans le deuxième modèle, et la présence des clés imaginaires [FK] ainsi que les clés primaires [PK], par contre dans le premier modèle on ne trouve que les clés primaires. On peut remarquer aussi que dans le deuxième modèle on trouve les types des attributs et la contrainte (NOT NULL), et leur absence dans le premier.

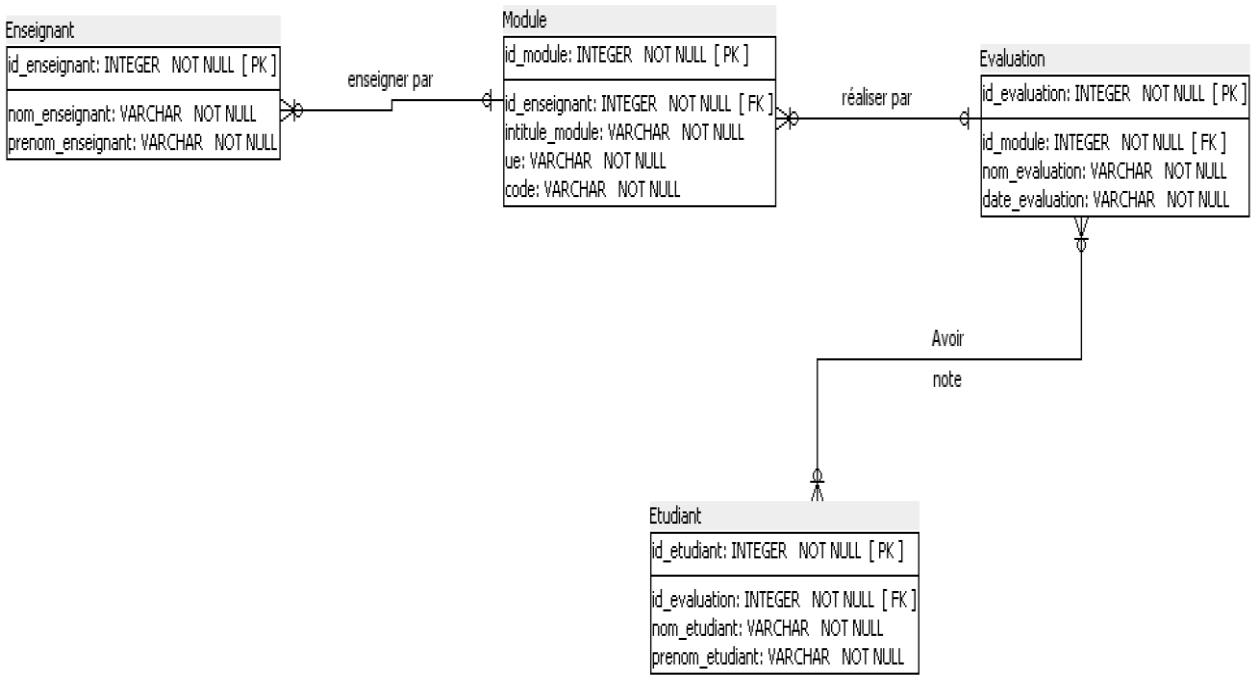
2. Association maillée cours/AGL :





⇨ On a deux associations maillées, la première faite manuellement sur DIA, et la deuxième avec l'AGL. On remarque l'absence des cardinalités dans le deuxième modèle, et la présence des clés imaginaires [FK] ainsi que les clés primaires [PK], par contre dans le premier modèle on ne trouve que les clés primaires. On peut remarquer aussi que dans le deuxième modèle on trouve les types des attributs et la contrainte (NOT NULL), et leur absence dans le premier.

3. Le schéma entités-associations avec l'AGL :



---

4. Le script SQL généré automatiquement par l'AGL :

```
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)
);
CREATE TABLE Module (
        id_module INTEGER NOT NULL,
        id_enseignant INTEGER NOT NULL,
        intitule_module VARCHAR NOT NULL,
        ue VARCHAR NOT NULL,
        code VARCHAR NOT NULL,
        CONSTRAINT id_module PRIMARY KEY (id_module)
);
CREATE TABLE Evaluation (
        id_evaluation INTEGER NOT NULL,
        id_module INTEGER NOT NULL,
        nom_evaluation VARCHAR NOT NULL,
        date_evaluation VARCHAR NOT NULL,
        CONSTRAINT id_evaluation PRIMARY KEY (id_evaluation)
```

```
);

CREATE TABLE Etudiant (
        id_etudiant INTEGER NOT NULL,
        id_evaluation INTEGER NOT NULL,
        nom_etudiant VARCHAR NOT NULL,
        prenom_etudiant VARCHAR NOT NULL,
        CONSTRAINT id_etudiant PRIMARY KEY (id_etudiant)
);
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 Module ADD CONSTRAINT module_module_fk
FOREIGN KEY (Parent_id_module)
REFERENCES Module (id_module)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
ALTER TABLE Evaluation ADD CONSTRAINT module_evaluation_fk
FOREIGN KEY (id_module)
REFERENCES Module (id_module)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
ALTER TABLE Etudiant ADD CONSTRAINT evaluation_etudiant_fk
FOREIGN KEY (id_evaluation)
REFERENCES Evaluation (id_evaluation)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
ALTER TABLE Etudiant ADD CONSTRAINT etudiant_etudiant_fk
FOREIGN KEY (Parent_id_etudiant)
REFERENCES Etudiant (id_etudiant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
DEFERRABLE INITIALLY DEFERRED;
```

5. On remarque que le script automatique est presque le même que celui fait manuellement, la seule différence c'est que dans le script automatique on trouve la mention ALTER TABLE qui permet de modifier les tables de plusieurs façon, avec les mentions ON DELETE et ON UPDATE ou encore ADD CONSTRAINT.

III. Peuplement des tables et requêtes

1. Le script de peuplement :

J'ai utilisé la requête COPY pour remplir les tables :

COPY Enseignant (id_enseignant, nom_enseignant, prenom_enseignant) FROM 'data.csv' DELIMITER ';' CSV HEADER;

COPY Module (id_module, code, ue, intitule_module, id_enseignant) FROM 'data.csv' DELIMITER ';' CSV HEADER;

COPY Evaluation (id_evaluation, nom_evaluation, date_evaluation, id_module) FROM 'data.csv' DELIMITER ';' CSV HEADER;

COPY Etudiant (id_etudiant, nom_etudiant, prenom_etudiant) FROM 'data.csv' DELIMITER ';' CSV HEADER;

COPY Avoir (id_etudiant, id_evaluation, note) FROM 'data.csv' DELIMITER ';' CSV HEADER;

2. Présentation de deux requêtes intéressantes sur la base de données :

- Cette requête donne les noms des évaluations qui étaient réalisées le 27/10/2021 :

  SELECT nom_evaluation FROM Evaluation WHERE date_evaluation = 27/10/2021 ;

- Cette requête donne les identifiants des étudiants et leurs classements dans l'évaluation ' Evaluation phase 2' dans l'ordre croissant

  SELECT note, id_etudiant FROM Avoir JOIN Evaluation ON Avoir.id_evaluation=Evaluation.id_evaluation WHERE nom_Evaluation='Evaluation phase 2' ORDER BY Avoir.note DESC ;