BUT1 Informatique
Bases de données et langage SQL

Rapport SAE S104

Réalisé par : Lina MEDANI

Groupe : SHANGO

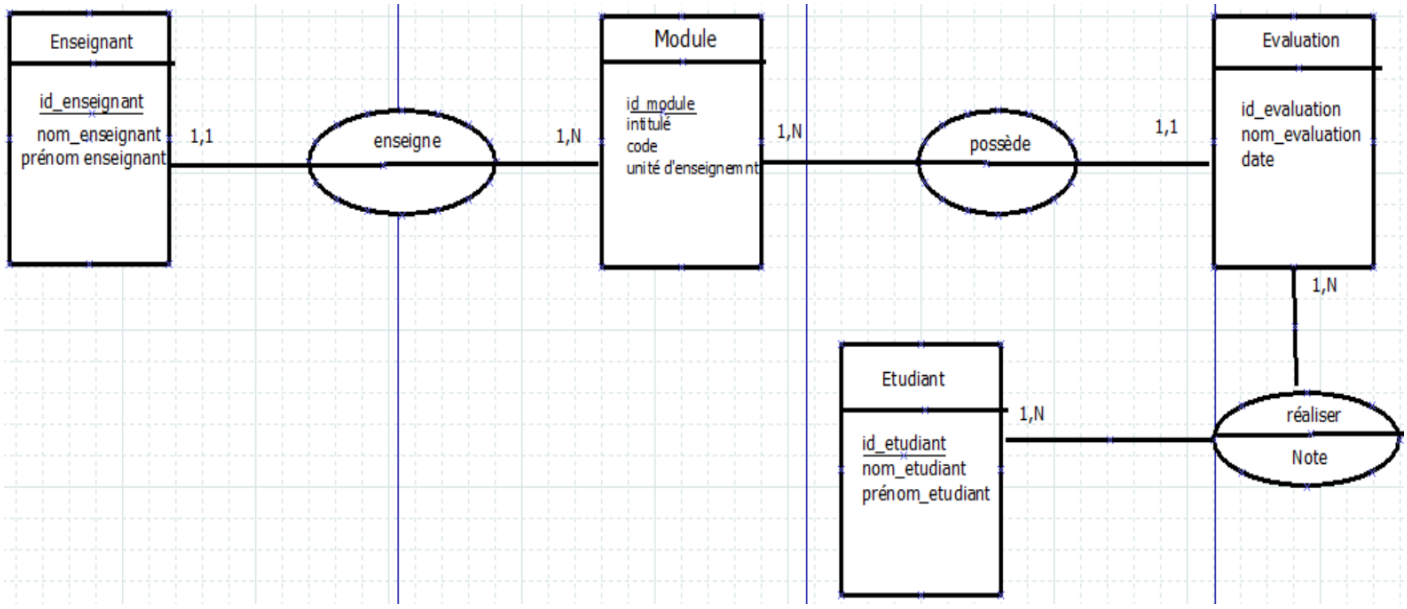
Enseignant : Laurent AUDIBERT

Année : 2023\ 2024

Sommaire :

Modélisation et script de création sans AGL

1-Modèle entités-associations :



2- Schéma Relationnel :

Enseignant(id_enseignant,nom_enseignant,prenom_enseignant,id_module)
Module(id_module,intitulé,code,unité d'enseignement)
Etudiant(id_etudiant,nom_etudiant,prénom_etudiant)
Evaluation(id_evaluation,nom_evaluation,date,id_module)
réaliser(id_etudiant,id_evaluation,note)

Avec la clé id_enseignant fait référence à la table enseignant, la clé id_module fait référence à la table module, la clé id_etudiant fait référence à la table étudiant, la clé id_evaluation fait référence à la table évaluation

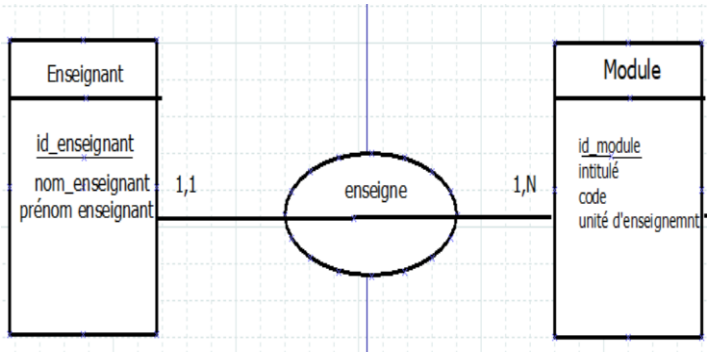
3-Script SQL de création des tables :

```
-- Database: SAE sql


-- DROP DATABASE IF EXISTS "SAE sql";


CREATE DATABASE "SAE sql"
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'French_France.1252'
    LC_CTYPE = 'French_France.1252'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;


    CREATE TABLE enseignant (
                        id_enseignant INTEGER PRIMARY KEY,
                        nom_enseignant VARCHAR,
                        prenom_enseignant VARCHAR,
                        id_module INTEGER REFERENCES modules
            );


    CREATE TABLE modules (
                        id_module INTEGER PRIMARY KEY,
                        intitulé_module VARCHAR,
                        Code VARCHAR,
,                       UE VARCHAR
            );
```

```
CREATE TABLE evaluation(
        id_evaluation INTEGER PRIMARY KEY,
        nom_evaluation VARCHAR,
        date_evaluation VARCHAR
    );

CREATE TABLE etudiant(
        id_etudiant INTEGER PRIMARY KEY,
        nom_etudiant VARCHAR,
        prenom_etudiant VARCHAR
    );
CREATE TABLE réaliser(
        id_etudiant INTEGER REFERENCES etudiant,
        id_evaluation INTEGER REFERENCES evaluation,
        note REAL
    );
```

Modélisation et script de création avec AGL

1-Illustrations comparatives cours/AGL association fonctionnelle :

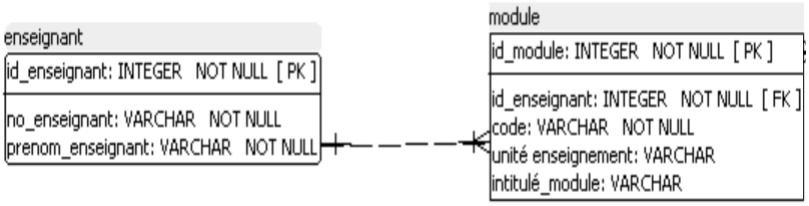


Modèle du cours association fonctionnelle

-Dans ce cas l'association et la cardinalité sont représentées entre les deux entités.

-Les cardinalités nous permettent de reconnaitre l'association fonctionnelle et faire le schéma relationnel en rajoutant la clé primaire de la relation module à la relation enseignant (coté d'entité ou la cardinalité maximale est 1).

-On remarque aussi dans le modèle entités-associations les clés étrangères sont absentes pour les apparaitre il faut faire le schéma relationnel.



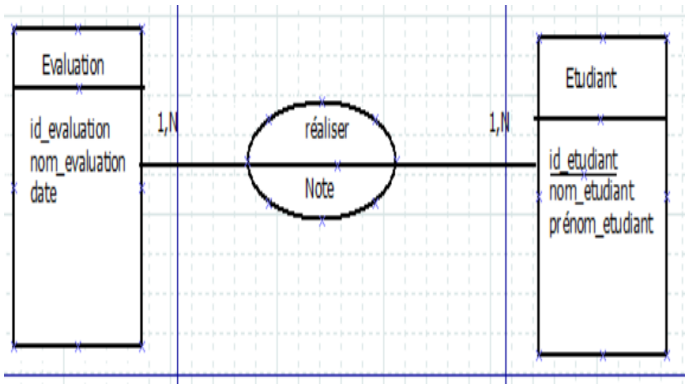
Modèle AGL association fonctionnelle

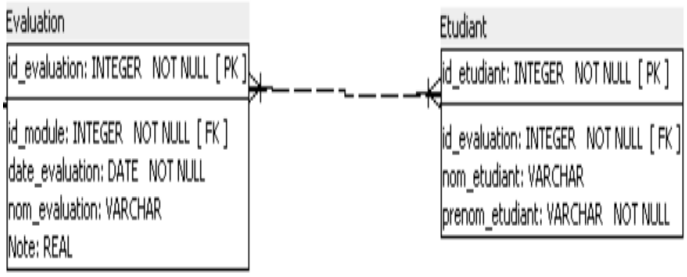
-Dans ce cas l'association et la cardinalité entre les deux associations ne sont pas représentés.

-Vu que la cardinalité est absente, la flèche entre les deux entités nous permet de reconnaitre la cardinalité tel que le trait à coté de l'entité enseignant indique que la cardinalité est exactement 1, et le trait avec la flèche à traits multiples à coté de l'entité module indique que la cardinalité minimale vaut 1, la maximale vaut N.

-On constate que les clés étrangères dans ce modèle AGL sont représentées dans les entités.

2-Illustrations comparatives cours/AGL association maillée:





Modèle association maillée cours :

-Dans ce modèle les associations et les cardinalités sont représentés, pour faire schéma relationnel on doit rajouter pour type association les deux clés primaires des deux entités et bien sûr les attributs de l'association aussi.

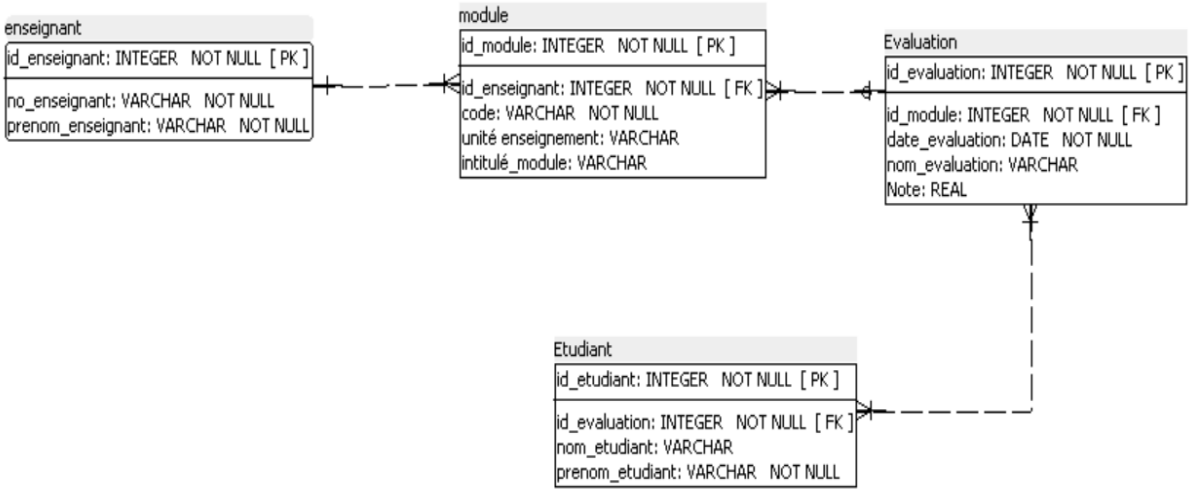
-L'absence des clés étrangères dans ce modèle.

Modèle association maillée AGL :

-Dans ce modèle les associations et les cardinalités ne sont pas représentés, mais on peut reconnaitre les cardinalités avec la flèche qui relie les deux entités tel que le trait indique que la cardinalité minimale vaut 1 et la flèche à traits multiples présente la cardinalité maximale qui vaut N.

-Les clés étrangères sont représentées dans les entités.

3-Modèle entités-associations réalisé par l'AGL :



4-Script SQL de création des tables généré automatiquement par 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,
        code VARCHAR NOT NULL,
        unité enseignement VARCHAR,
        intitulé_module VARCHAR,
        CONSTRAINT id_module PRIMARY KEY (id_module)
);
CREATE TABLE Evaluation (
        id_evaluation INTEGER NOT NULL,
        id_module INTEGER NOT NULL,
        id_enseignant INTEGER NOT NULL,
        date_evaluation DATE NOT NULL,
        nom_evaluation VARCHAR,
        Note REAL,
        CONSTRAINT id_evaluation PRIMARY KEY (id_evaluation)
);
CREATE TABLE Etudiant (
        id_etudiant INTEGER NOT NULL,
        id_evaluation INTEGER NOT NULL,
        id_module INTEGER NOT NULL,
        id_enseignant INTEGER NOT NULL,
        nom_etudiant VARCHAR,
        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 Evaluation ADD CONSTRAINT module_evaluation_fk

FOREIGN KEY (id_enseignant, id_module)

REFERENCES module (id_enseignant, id_module)

ON DELETE NO ACTION

ON UPDATE NO ACTION

NOT DEFERRABLE;


ALTER TABLE Etudiant ADD CONSTRAINT evaluation_etudiant_fk

FOREIGN KEY (id_evaluation, id_module, id_enseignant)

REFERENCES Evaluation (id_evaluation, id_module, id_enseignant)

ON DELETE NO ACTION

ON UPDATE NO ACTION

NOT DEFERRABLE;
```

5-Les différences entre les scripts produits manuellement et automatiquement :

- Dans le script fait manuellement on écrit après « PRIMARY KEY » a coté de la colonne qui représente la clé primaire alors que dans le script automatique, il mentionne d'abord les différentes colonnes de la table ensuite il écrit « CONSTRAINT 'nom de la colonne qui identifie la clé primaire' ».
- Les clés étrangères dans les tables écrites manuellement sont mentionnées dans CREATE TABLE comme « id_module INTEGER REFERENCES modules », mais dans les tables produites

par l'AGL elles sont mentionnées après la création des tables dans « ALTER TABLE nom_table ADD CONSTRAINT clé étrangère_fk »

- Dans le script AGL Il y a des commentaires au-dessous des tables crées ON DELETE, ON UPDATE, ils indiquent les modifications faites sinon il écrit NO ACTION pour les deux.

Peuplement de tables des requêtes :

1-Les différentes étapes du script de peuplement :

```
CREATE TABLE data(
                    id_enseignant INTEGER,
                    nom_enseignant VARCHAR,
                    prénom_enseignant VARCHAR,
                    id_module INTEGER,
                    intitulé_module VARCHAR,
                    code VARCHAR,
                    UE VARCHAR,
                    id_evaluation INTEGER,
                    nom_evaluation VARCHAR,
                    date_evaluation VARCHAR,
                    id_etudiant INTEGER,
                    nom_etudiant VARCHAR,
                    prenom_etudiant VARCHAR,
                    note REAL) ;
Copy data from 'data.csv' WITH (Format CSV, HEADER, DELIMITER ';');
```

-D'abord, on crée la table data qui rassemble les colonnes de la base de données.

-Ensuite, avec « COPY » on copie toutes les données du fichier 'data.csv' pour remplir la table crée.

-Format correspond au format du fichier qui est csv.

-Header est utilisé pour supprimer la première ligne du fichier car elle contient l'intitulé des colonnes de la base qui sont déjà créés par la requête « CREATE TABLE ».

-DELEMITER sert à séparer avec le ' ; ' les lignes du fichier (data.csv).

2-Présentation commentée de deux requêtes intéressantes sur la base de données :

→Donner les noms, les prénoms et les notes des étudiants classées en ordre croissant dans le module de bases de données.

SELECT nom_etudiant, prenom_etudiant, note FROM data ORDER BY note ;

On sélectionne d'abord le nom, le prénom et la note des étudiants.

ORDER BY sert à ordonner les notes d'un ordre croissant.

→Quels sont les noms des évaluations passées dans chaque module ?

SELECT nom_evaluation FROM data GROUP BY intitulé_module ;

On sélectionne les évaluations avec SELECT et les regrouper par module avec « GROUP BY ».