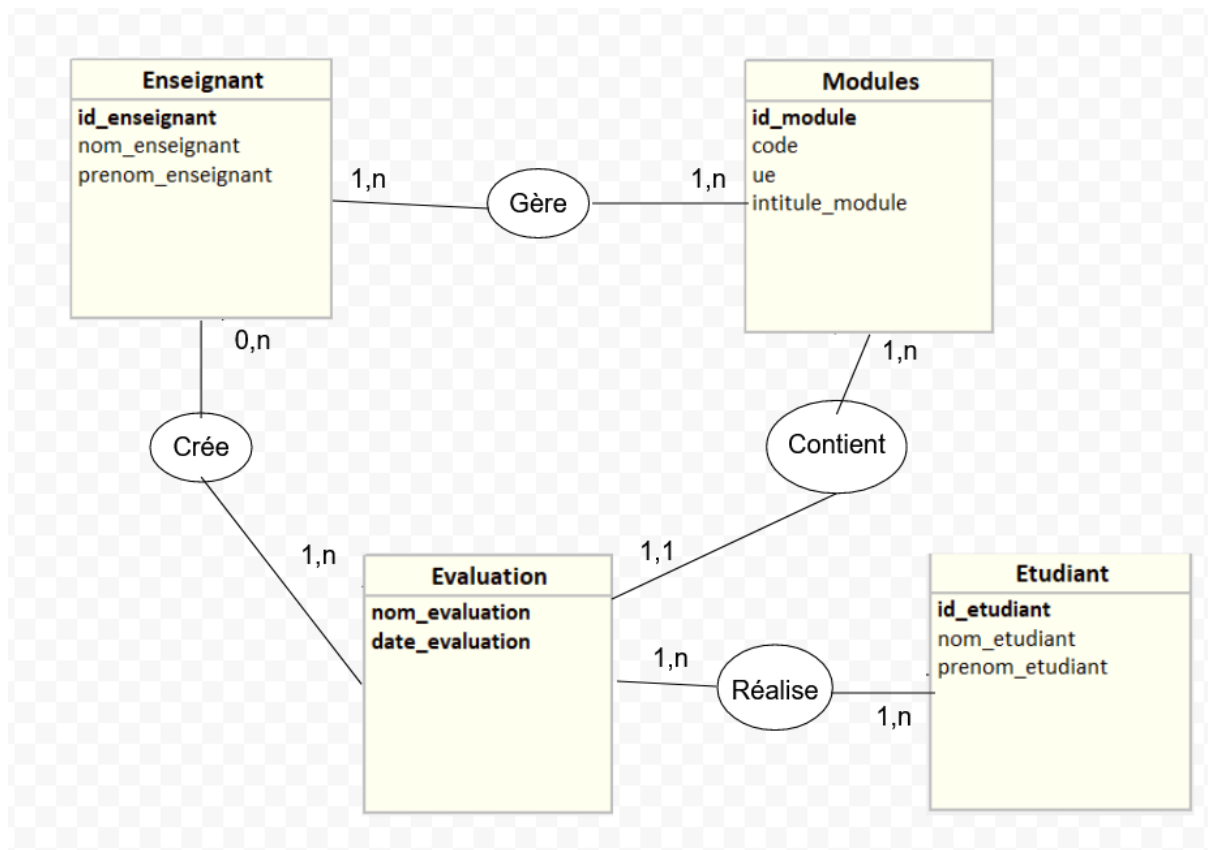


BELMESSAOUD Imran Zeus BUT1 INFO

SAE SQL

2.1 :

Schéma relationnel avec Google Drawings :



Script SQL manuel :

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

```
CREATE TABLE Modules(  
  id_module INTEGER PRIMARY KEY,  
  code VARCHAR NOT NULL,  
  ue VARCHAR NOT NULL,  
  intitule_module VARCHAR NOT NULL  
);
```

```
CREATE TABLE Etudiant(  
  id_etudiant INTEGER PRIMARY KEY,  
  nom_etudiant VARCHAR NOT NULL,
```

```

    prenom_etudiant VARCHAR NOT NULL
);

```

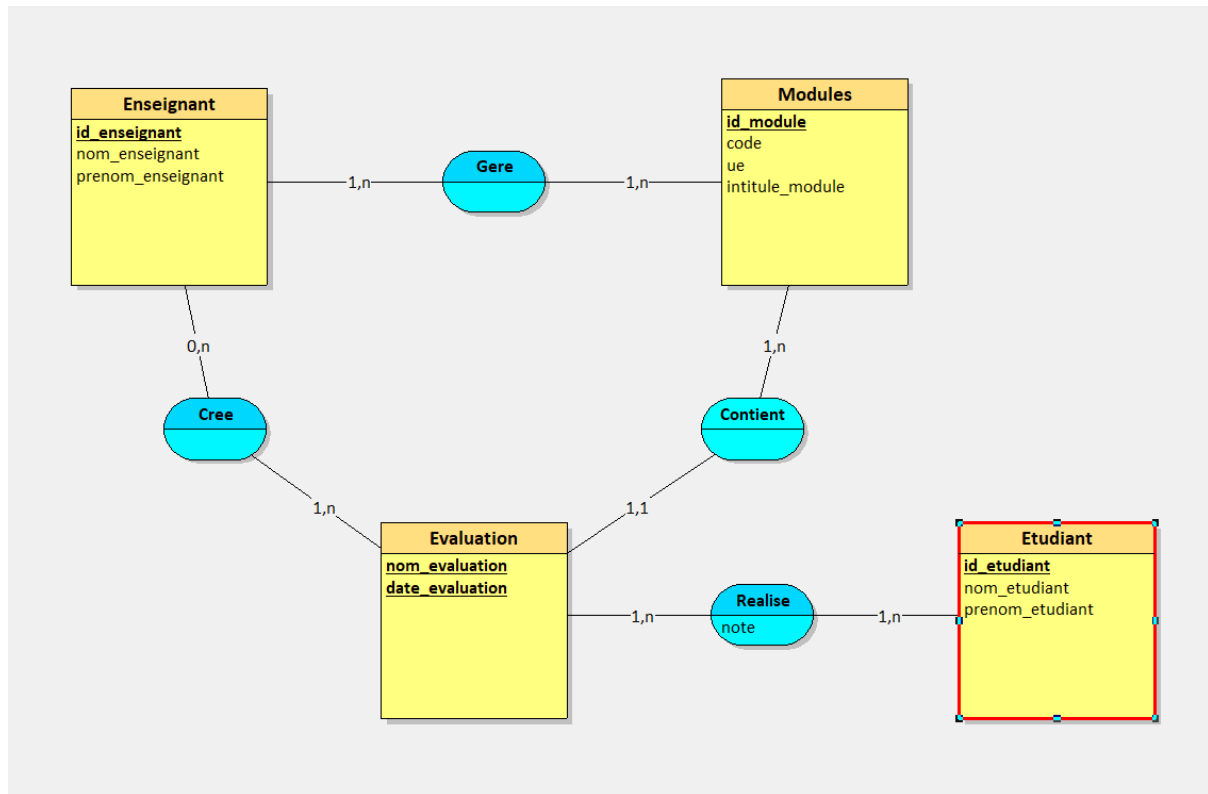
```

CREATE TABLE Evaluation(
    nom_evaluation VARCHAR NOT NULL,
    date_evaluation DATE
);

```

2.2 :

Modèle entités associations :



Script SQL généré automatiquement par l'AGL:

```

CREATE TABLE Enseignant(
    id_enseignant BIGINT,
    nom_enseignant VARCHAR(50),
    prenom_enseignant VARCHAR(50),
    PRIMARY KEY(id_enseignant)
);

```

```

CREATE TABLE Modules(
    id_module BIGINT,
    code VARCHAR(50),
    ue VARCHAR(50),
    intitule_module VARCHAR(50),
    PRIMARY KEY(id_module)
);

```

```
CREATE TABLE Etudiant(  
    id_etudiant BIGINT,  
    nom_etudiant VARCHAR(50),  
    prenom_etudiant VARCHAR(50),  
    PRIMARY KEY(id_etudiant)  
);
```

```
CREATE TABLE Evaluation(  
    nom_evaluation VARCHAR(50),  
    date_evaluation DATE,  
    id_module BIGINT NOT NULL,  
    PRIMARY KEY(nom_evaluation, date_evaluation),  
    FOREIGN KEY(id_module) REFERENCES Modules(id_module)  
);
```

```
CREATE TABLE Cree(  
    id_enseignant BIGINT,  
    nom_evaluation VARCHAR(50),  
    date_evaluation DATE,  
    PRIMARY KEY(id_enseignant, nom_evaluation, date_evaluation),  
    FOREIGN KEY(id_enseignant) REFERENCES Enseignant(id_enseignant),  
    FOREIGN KEY(nom_evaluation, date_evaluation) REFERENCES  
Evaluation(nom_evaluation, date_evaluation)  
);
```

```
CREATE TABLE Gere(  
    id_enseignant BIGINT,  
    id_module BIGINT,  
    PRIMARY KEY(id_enseignant, id_module),  
    FOREIGN KEY(id_enseignant) REFERENCES Enseignant(id_enseignant),  
    FOREIGN KEY(id_module) REFERENCES Modules(id_module)  
);
```

```
CREATE TABLE Realise(  
    nom_evaluation VARCHAR(50),  
    date_evaluation DATE,  
    id_etudiant BIGINT,  
    note BIGINT,  
    PRIMARY KEY(nom_evaluation, date_evaluation, id_etudiant),  
    FOREIGN KEY(nom_evaluation, date_evaluation) REFERENCES  
Evaluation(nom_evaluation, date_evaluation),  
    FOREIGN KEY(id_etudiant) REFERENCES Etudiant(id_etudiant)  
);
```

Discussion différence entre les deux scripts SQL :

Il y a plusieurs différences entre les deux scripts SQL fournis par l'AGL et manuel:

Dans le premier script, les clés primaires sont définies explicitement pour chaque table en utilisant la syntaxe "PRIMARY KEY(champ)". Dans le deuxième script, les clés primaires sont définies en utilisant la syntaxe "INTEGER PRIMARY KEY" pour les champs appropriés.

Dans le premier script, les contraintes de clé étrangère sont définies explicitement pour les tables "Evaluation", "Cree" et "Realise" en utilisant la syntaxe "FOREIGN KEY(champ) REFERENCES table(champ)".

Dans le premier script, les tables "Cree" et "Gere" contiennent des champs supplémentaires pour gérer les relations entre les enseignants, les modules et les évaluations et les propriétés de certaines colonnes, comme le type et la longueur, sont explicitement définies pour chaque colonne.

Dans le deuxième script, certaines propriétés de colonne, les contraintes de clé étrangère et les relations ne sont pas explicitement définies.

Dans le premier script, certaines colonnes ont des contraintes de NOT NULL, dans le deuxième script ces contraintes ne sont pas définies.

Les scripts générés sont alors totalement différents même si l'on veut obtenir le même but final.

2.3 :

Script de peuplement des tables :

Dans ce script, on créera une table temporaire, puis on entrera la commande permettant de copier les colonnes de tables à partir d'un fichier t et les délimiter par des point-virgules pour coller chaque colonne dans la table temporaire et enfin on finira par copier chaque colonne de la table temporaire dans chacune des colonnes de la vraie base de données :

```
CREATE TABLE temporaire(  
    id_enseignant INTEGER,  
    nom_enseignant VARCHAR NOT NULL,  
    prenom_enseignant VARCHAR NOT NULL  
    id_module INTEGER PRIMARY KEY,  
    code VARCHAR NOT NULL,  
    ue VARCHAR NOT NULL,  
    intitule_module VARCHAR NOT NULL  
    id_etudiant INTEGER PRIMARY KEY,  
    nom_etudiant VARCHAR NOT NULL,  
    prenom_etudiant VARCHAR NOT NULL  
    nom_evaluation VARCHAR NOT NULL,  
    date_evaluation DATE,  
    note REAL  
);
```

```
\copy temporaire from 'data.csv' delimiter ';' csv header;
```

```
INSERT INTO etudiant
```

```
SELECT DISTINCT
```

```
id_enseignant
```

```
nom_enseignant
```

```
prenom_enseignant FROM temporaire ORDER BY id_enseignant
```

```
SELECT DISTINCT
```

```
id_module
```

```
code
```

```
ue
```

```
intitule_module FROM temporaire ORDER BY id_module
```

```
SELECT DISTINCT
```

```
id_etudiant
```

```
nom_etudiant
```

```
prenom_etudiant FROM temporaire ORDER BY id_etudiant
```

```
SELECT DISTINCT
```

```
nom_evaluation
```

```
date_evaluation
```

```
note FROM temporaire ORDER BY nom_evaluation
```