Berdah **SAE S104.**

Clément

Zeus

**Rapport**

° 2.1 - Modélisation et script de création “ sans AGL “.

* Schéma relationnel.
* Script SQL de création de tables.

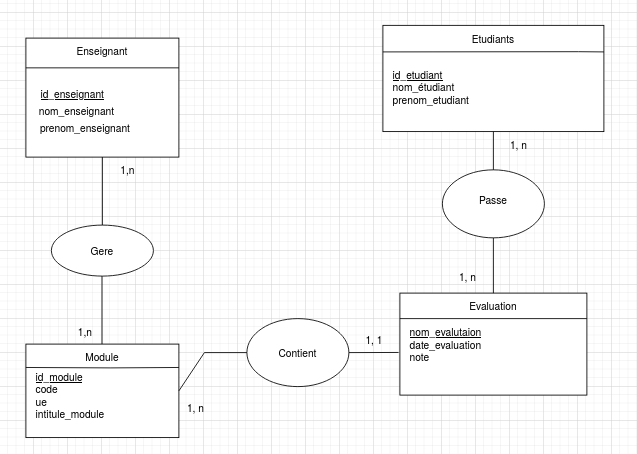
° 2.2 - Modélisation et script de création “ avec AGL “.

* Illustrations comparatives cours/AGL commentées d’une association fonctionnelle.
* Illustrations comparatives cours/AGL commentées d’une association maillée.
* Modèle EA réalisé avec l’AGL
* Script SQL de création des tables généré automatiquement par l’AGL
* Discussion sur les différences entre les scripts produits manuellement et automatiquement (AGL)

° 2.3 - Peuplement des tables et requêtes.

* Description commentée des différentes étapes de votre script de peuplement.
* Présentation commentée de deux requêtes intéressantes sur la base de données.

**° 2.1 - Modélisation entités-associations (EA) respectant la syntaxe du cours.**



2.1 - Schéma relationnel.

ENSEIGNANT(id\_enseignant, nom\_enseignant, prenom\_enseignant)

→ GERE(id\_module, id\_enseignant)

MODULE(id\_module, code, ue, intitule\_module)

EVALUATION(nom\_evaluation, date\_evalutaion, note)

→ PASSE(nom\_evalutaion, id\_etudiant)

ETUDIANT(id\_etudiant, nom\_etudiant, prenom\_etudiant)

2.1 - Script SQL de création de tables.

CREATE TABLE Enseignant (

id\_enseignant INTEGER PRIMARY KEY,

nom\_enseignant VARCHAR (100),

prenom\_enseignant VARCHAR (100)

) ;

CREATE TABLE Module (

id\_module INTEGER PRIMARY KEY,

code VARCHAR (4),

ue VARCHAR (4),

intitule\_module VARCHAR (100)

) ;

CREATE TABLE Evaluation (

nom\_evaluation VARCHAR (100) PRIMARY KEY,

date\_evaluation DATE,

note DOUBLE PRECISION

) ;

CREATE TABLE Etudiant (

id\_etudiant INTEGER PRIMARY KEY,

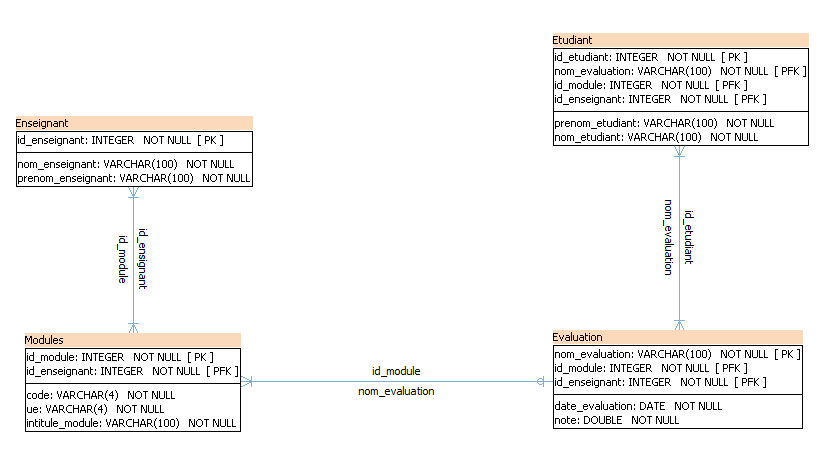
nom\_etudiant VARCHAR (100),

prenom\_etudiant VARCHAR (100)

) ;

**° 2.2 - Modélisation et script de création “ avec AGL “.**

2.2.3 - Modèle EA réalisé avec l’AGL



2.2.4 - Script SQL de création des tables généré automatiquement par l’AGL

CREATE TABLE Enseignant (

id\_enseignant INTEGER NOT NULL,

nom\_enseignant VARCHAR(100) NOT NULL,

prenom\_enseignant VARCHAR(100) NOT NULL,

CONSTRAINT id\_enseignant PRIMARY KEY (id\_enseignant)

);

CREATE TABLE Modules (

id\_module INTEGER NOT NULL,

id\_enseignant INTEGER NOT NULL,

code VARCHAR(4) NOT NULL,

ue VARCHAR(4) NOT NULL,

intitule\_module VARCHAR(100) NOT NULL,

CONSTRAINT id\_module PRIMARY KEY (id\_module, id\_enseignant)

);

CREATE TABLE Evaluation (

nom\_evaluation VARCHAR(100) NOT NULL,

id\_module INTEGER NOT NULL,

id\_enseignant INTEGER NOT NULL,

date\_evaluation DATE NOT NULL,

note DOUBLE NOT NULL,

CONSTRAINT nom\_evaluation PRIMARY KEY (nom\_evaluation, id\_module, id\_enseignant)

);

CREATE TABLE Etudiant (

id\_etudiant INTEGER NOT NULL,

nom\_evaluation VARCHAR(100) NOT NULL,

id\_module INTEGER NOT NULL,

id\_enseignant INTEGER NOT NULL,

prenom\_etudiant VARCHAR(100) NOT NULL,

nom\_etudiant VARCHAR(100) NOT NULL,

CONSTRAINT id\_etudiant PRIMARY KEY (id\_etudiant, nom\_evaluation, id\_module, id\_enseignant)

);

2.2.5 - Discussion sur les différences entre les scripts produits manuellement et automatiquement (AGL)

Au niveau de l'écriture entre ces deux scripts, on peut noter plusieurs différences :

* Tout d'abord le script généré automatiquement a défini chaque colonne comme étant non nulles (NOT NULL).
* Il y a également une différence au niveau de l'illustration des clés primaires par rapport aux colonnes d'autres tables avec qui elles compositent.
* Il y a aussi des références clés étrangères qui ne sont pas présentes dans le script manuel, qui permettent de créer des relations entre les tables.

Le script automatique est donc plus détaillé et plus complet qu’un script écrit manuellement.

2.3.1 - Description commentée des différentes étapes de votre script de peuplement.

Voici le script de peuplement :

SET datestyle = dmy;

CREATE TABLE temp(

id\_enseignant INTEGER,

nom\_enseignant VARCHAR,

prenom\_enseignant VARCHAR,

id\_module INTEGER,

code VARCHAR,

ue VARCHAR,

intitule\_module VARCHAR,

nom\_evaluation VARCHAR,

date\_evaluation DATE,

note REAL,

id\_etudiant INTEGER,

nom\_etudiant VARCHAR,

prenom\_etudiant VARCHAR

) ;