

SAE Bases de Données et langage SQL S104

id_enseignant	nom_enseignant	prenom_enseignant	id_module	code	ue	intitule_moc	nom_evalua	date_evalua	note	id_etudiant	nom_etudiant	prenom_etudiant
145	Heron	Anne	2	R101	UE12	Initiation au	Contrôle mo	27/10/2021	6	124	Calamel	Claudius
161	Coignard	Charles	18	R107	UE12	Outils math	Contrôle le 1	24/11/2021	7.25	37	Havez	Catherine
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle 3 L	08/11/2021	6	7	Schilling	Marius
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle le En	20/10/2021	12.16	51	Canet	Georgette
150	Gervais	Vincent	7	R110	UE12	Anglais tech	Oral present	20/01/2022	15.5	126	Chaumaz	Herve
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle 2 R	04/10/2021	10	43	Bagur	Gerard
145	Heron	Anne	2	R101	UE12	Initiation au	Contrôle mo	29/09/2021	17.75	63	Carron	Oceane
144	Helin	Mohamed	1	S104	UE13	Cr	ation d Evaluation p	06/09/2021	2.3	131	Delayen	Cindy
146	Denis	Olivier	3	R102	UE12	D	veloppement Contrôle le co	15/12/2021	13	123	Badji	Elodie
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle Mini-test 2 R	04/10/2021	9.5	115	Dangreux	Ngoc
144	Helin	Mohamed	1	S104	UE13	Cr	ation d Evaluation p	06/09/2021	8.3	6	Franceschi	Bruno
145	Heron	Anne	2	R101	UE12	Initiation au	Minicontrolé	20/09/2021	1.5	20	Tessier	Matthieu
159	Carrere	Mohamed	16	R105	UE12	Introduction	Contrôle le co	13/10/2021	16.25	85	Guiard	Alphonse
149	Martos	Marcelle	6	R111	UE12	Bases de la c	CV (androme	22/01/2022	10	120	Abdennebi	Joseph
156	Lusseau	Patrice	13	S105	UE13	Recueil de b	Projet maria	18/01/2022	12	90	Blot	Gabrielle
144	Helin	Mohamed	1	S104	UE13	Cr	ation d Evaluation p	06/09/2021	9	110	Frejafon	Antonio
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle le En	20/10/2021	13.44	129	Vallat	Stefan
148	Sabatier	Michele	5	R103	UE12	Introduction	QCM	21/01/2022	11.5	20	Tessier	Matthieu
156	Lusseau	Patrice	13	S105	UE13	Recueil de b	Projet maria	18/01/2022	20	98	Etcheverry	Therese
152	Grosperin	Yvon	9	S106	UE13	D	couvert Note globale	16/01/2022	15	123	Badji	Elodie
161	Coignard	Charles	18	R107	UE12	Outils math	Contrôle le 1	24/11/2021	14.5	5	Guillet	Jerome
145	Heron	Anne	2	R101	UE12	Initiation au	Minicontrolé	12/10/2021	4.75	89	Boumahdi	Paul
155	Grandin	Antoine	12	R112	UE12	Projet prof:	Expos	04/01/2022	17	3	Faity	Olivier
160	Montier	Joao	17	R109	UE12	É	conomie Examen fina	10/11/2021	15	110	Frejafon	Antonio
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle Mini-test Pr	22/11/2021	14.5	92	Louviot	Jacques
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle le lo	08/12/2021	10	133	Dufour	Paul
145	Heron	Anne	2	R101	UE12	Initiation au	Contrôle le 4	21/01/2022	2	50	Bori	Josephine
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle Mini-test Pr	22/11/2021	15.5	120	Abdennebi	Joseph
145	Heron	Anne	2	R101	UE12	Initiation au	Minicontrolé	13/09/2021	2.25	41	Casagrande	Gaston
161	Coignard	Charles	18	R107	UE12	Outils math	Contrôle le 2	12/01/2022	5.75	61	Adam	Aime
151	Leroy	Vincent	8	R108	UE12	Gestion de p	Examen fina	20/01/2022	13.46	92	Louviot	Jacques
157	Donizeau	Leon	14	R106	UE12	Math	Contrôle Mini-test 2 R	04/10/2021	9.5	123	Badji	Elodie
147	Selosse	Frederic	4	S102	UE13	Comparaisor	Evaluation ci	17/01/2022	5.8	6	Franceschi	Bruno
153	Caplot	Prosper	10	S103	UE13	Installation	Contrôle valuation	13/12/2021	16	56	Schmitt	Geraldine
150	Gervais	Vincent	7	R110	UE12	Anglais tech	Oral present	20/01/2022	12.5	44	Ravallier	Felix

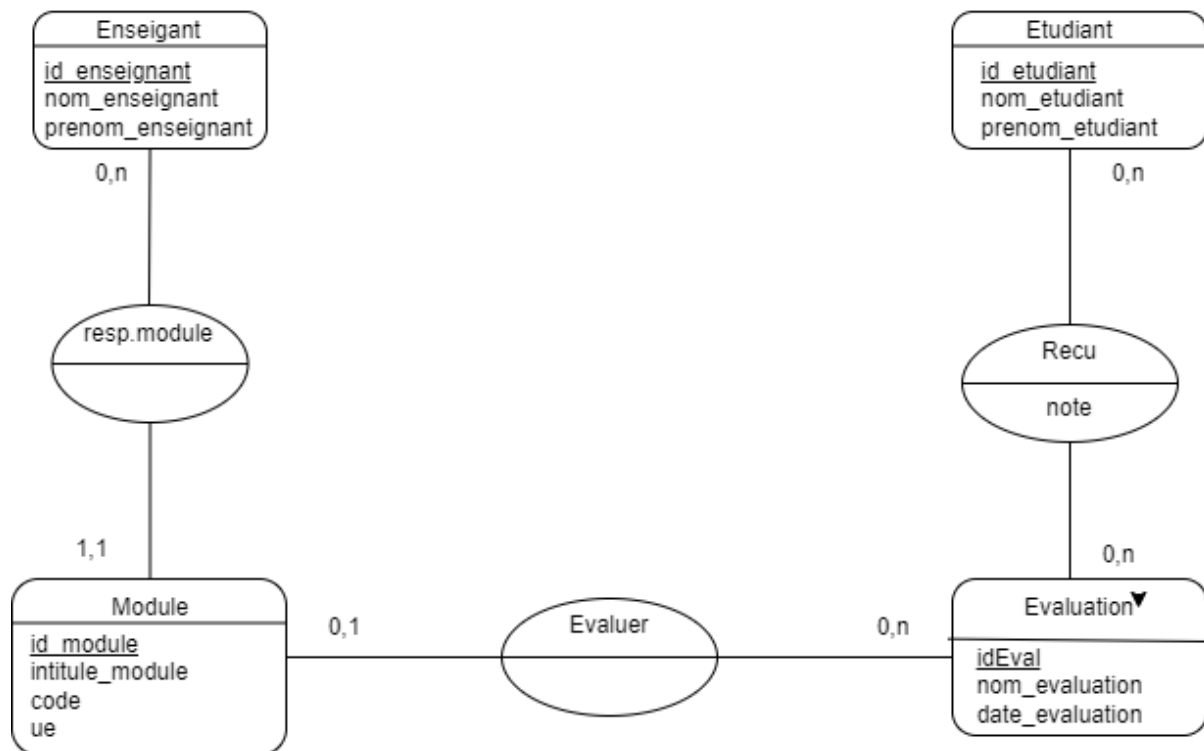
I.Modélisation et script de création sans AGL

II.Modélisation et script de création avec AGL

III.Peuplement des tables et requêtes

I.Modélisation et script de création sans AGL

1.Modèle entités-associations respectant la syntaxe du cours :



2.Schéma relationnel :

Etudiant(id_etudiant, nom_etudiant, prenom_etudiant)
Enseignant(id_enseignant, nom_enseignant, prenom_enseignant)
Module(id_module, intitulé_module, code, ue)
resp.module(id_enseignant , id_module)
Evaluation (nom_evaluation , date_evaluation)
Evaluer(idModule, nom_evaluation)
reçu(nom_evaluation, id_etudiant, note)

3.Script SQL de création des tables :

```
CREATE TABLE Etudiant(  
id_etudiant INTEGER Primary KEY,  
nom_etudiant VARCHAR NOT NULL,  
prenom_etudiant VARCHAR NOT NULL);
```

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

```
CREATE TABLE Evaluation(  
nom_evaluation VARCHAR Primary KEY,  
date_evaluation DATE);
```

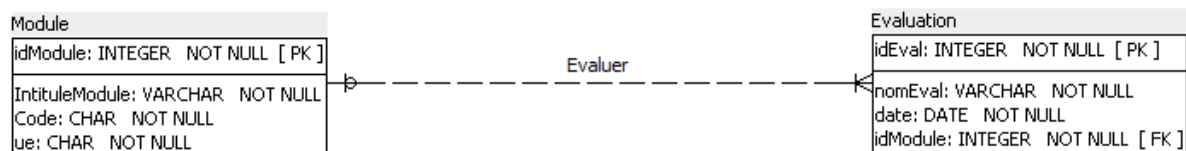
```
CREATE TABLE Module(  
id_module INTEGER Primary KEY,  
intitule_Module VARCHAR NOT NULL,  
code CHAR NOT NULL,  
ue CHAR NOT NULL);
```

```
CREATE TABLE resp_module(  
id_enseignant INTEGER REFERENCES Enseignant,  
id_module INTEGER REFERENCES Module,  
Primary KEY(id_module,id_enseignant));
```

```
CREATE TABLE Recu(  
id_eval INTEGER REFERENCE Evaluation,  
id_etudiant INTEGER REFERENCES Etudiant,  
note REAL CHECK(note > 0 OR note IS NULL);
```

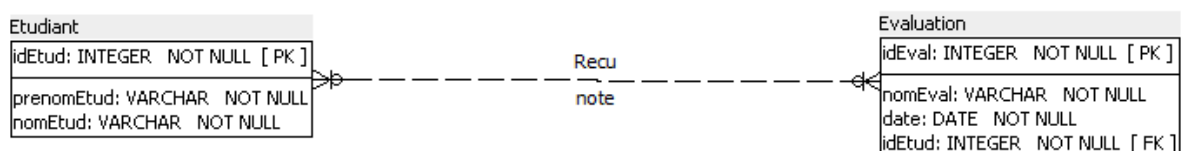
II.Modélisation et script de création avec AGL

1.Illustrations comparatives cours/AGL d'une association fonctionnelle



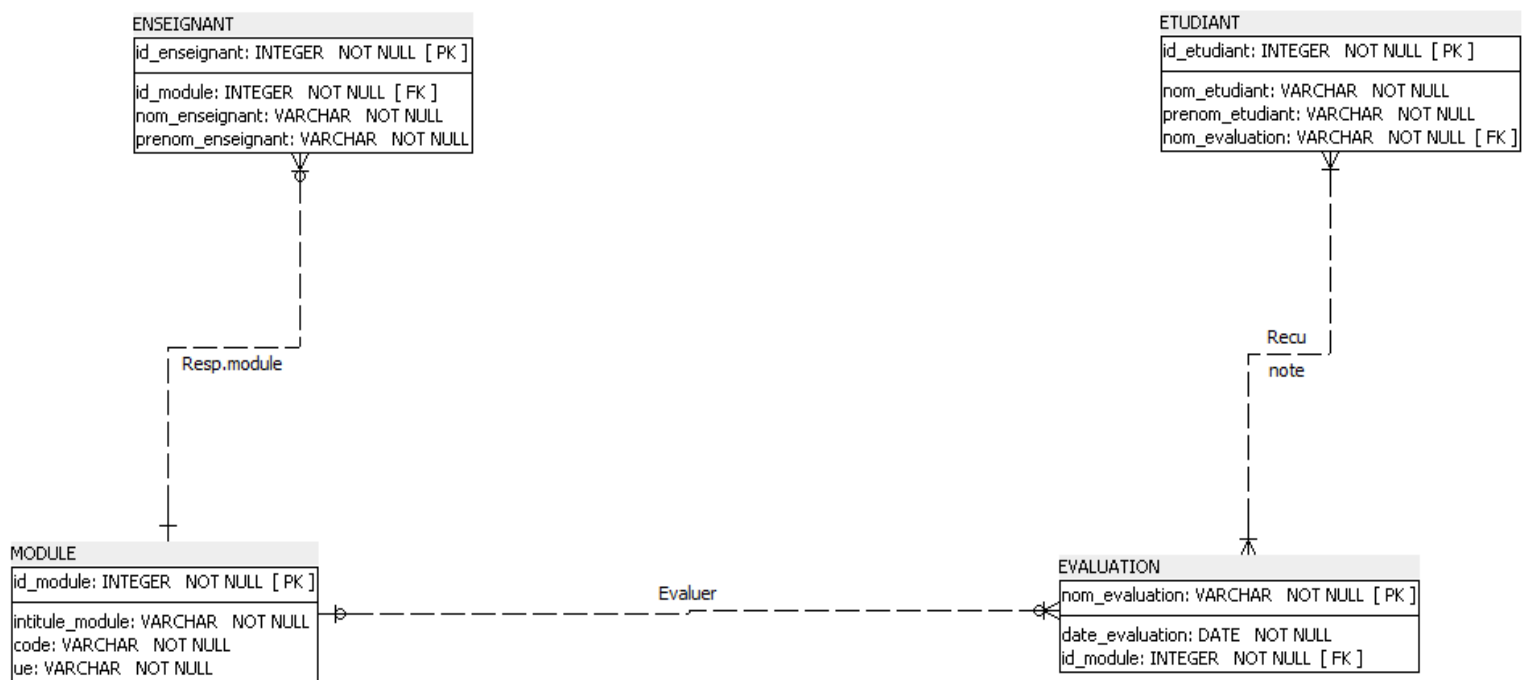
Pour l'association fonctionnelle les cardinalité change elle sont désigné par <, |, ° et la ligne est sous forme de pointillé Il y a aussi la clé idModule qui se crée en bas de Evaluation.

2.Illustrations comparatives cours/AGL d'une association maillée



Pour l'association maillée les cardinalité change elles sont désigné par des <, |, ° et la ligne est sous forme de pointillé. Il y a aussi la clé idEtud qui se crée en bas de Evaluation.

3.Modèle entités-association avec l'AGL:



4.Script SQL avec AGL:

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

```
CREATE TABLE EVALUATION (  
    nom_evaluation VARCHAR NOT NULL [PK],  
    date_evaluation DATE NOT NULL,  
    id_module INTEGER NOT NULL [FK],  
    CONSTRAINT nom_evaluation PRIMARY KEY (nom_evaluation)
```

```
        nom_evaluation VARCHAR NOT NULL,  
        date_evaluation DATE NOT NULL,  
        id_module INTEGER NOT NULL,  
        CONSTRAINT nom_evaluation PRIMARY KEY  
(nom_evaluation)  
);
```

```
CREATE TABLE ETUDIANT (  
        id_etudiant INTEGER NOT NULL,  
        nom_etudiant VARCHAR NOT NULL,  
        prenom_etudiant VARCHAR NOT NULL,  
        nom_evaluation VARCHAR NOT NULL,  
        CONSTRAINT id_etudiant PRIMARY KEY (id_etudiant)  
);
```

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

```
ALTER TABLE ENSEIGNANT ADD CONSTRAINT MODULE_ENSEIGNANT_fk  
FOREIGN KEY (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 (nom_evaluation)  
REFERENCES EVALUATION (nom_evaluation)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

III. Peuplement des tables et requêtes

1. Description commentée des différentes étapes de votre script de peuplement

Les différences se trouvent principalement sur l'ordre de création des tables, des différentes commandes et associations. Les clés étrangères sont désignées par fk et les clés primaires par pk. Ils utilisent des Alter table pour pouvoir créer dans un premier temps les tables puis les spécifier grâce à l'Alter table on peut y retrouver les clés étrangères.

Peuplement des tables:

J'ai utilisé pgadmin. Dans un premier temps j'ai créé une table pour pouvoir stocker le fichier csv. Cette table contient chaque attribut du csv avec leur bon type, j'ai utilisé le système de création de pgadmin qui me permet d'utiliser leur syntaxe par exemple les VARCHAR étaient appelés character varying.

```
CREATE TABLE tablestockcsv  
(  
    id_enseignant integer,  
    nom_enseignant character varying,  
    prenom_enseignant character varying,  
    id_module integer,  
    code character varying,  
    ue character varying,  
    intitule_module character varying,  
    nom_evaluation character varying,  
    date_evaluation date,  
    note numeric,  
    id_etudiant integer,  
    nom_etudiant character varying,  
    prenom_etudiant character varying  
);
```

Une fois ma table "tablestockcsv" créée j'ai utilisé:

```
COPY tablestockcsv FROM 'D:\AGL\data.csv' DELIMITER ';' CSV Header;
```

pour la remplir, J'ai ensuite implémenté mes table une par une dans le bonne ordre et à chaque table créée je la remplissais grâce à la commande:

```
INSERT INTO nomdelatable(SELECT DISTINCT id,attr1,attr2... FROM  
tablestockcsv);
```

J'ai procédé ainsi pour pouvoir cibler la table ou un problème serait survenu. Une fois toutes les création de table et insert fait le peuplement était finie.

Voici toute ma table final :

```
CREATE TABLE Etudiant(  
id_etudiant INTEGER Primary KEY,  
nom_etudiant VARCHAR NOT NULL,  
prenom_etudiant VARCHAR NOT NULL);
```

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

```
CREATE TABLE Evaluation(  
id_eval INTEGER Primary Key  
nom_evaluation VARCHAR NOT Null,  
date_evaluation DATE,  
id_Module INTEGER REFERENCES Module);
```

```
CREATE TABLE Module(  
id_module INTEGER Primary KEY,  
id_respmodule INTEGER REFERENCES Enseignant(id_enseignant),  
intitule_Module VARCHAR NOT NULL,  
code CHAR NOT NULL,  
ue CHAR NOT NULL);
```

```
CREATE TABLE Recu(  
id_etudiant INTEGER REFERENCES Etudiants,  
id_Evaluation INTEGER REFERENCES Evaluation,  
note REAL CHECK(note > 0 OR note IS NULL),  
Primary KEY(id_etudiant,id_Evaluation));
```

2. Présentation commenté de deux requêtes intéressantes sur le base de données

Requête:

<pre>SELECT DISTINCT Etudiant.nom_etudiant, Etudiant.prenom_etudiant From Etudiant JOIN Recu On Etudiant.id_etudiant = Recu.id_etudiant WHERE Recu.note<10;</pre>	<p>Pour voir les étudiants qui ont eu en dessous de la moyenne. La requête car on peut gérer tous les individu qui ont eu en dessous de la moyenne</p>
<pre>SELECT Evaluation.date From Evaluation JOIN Module ON Evaluation.id_module = Module.id_module WHERE Module.intitule_module = " Créations d'une base de données"</pre>	<p>Les dates d'évaluation du module création d'une base de données. La requête est intéressante car on peut observer une jointure et plusieurs conditions.</p>