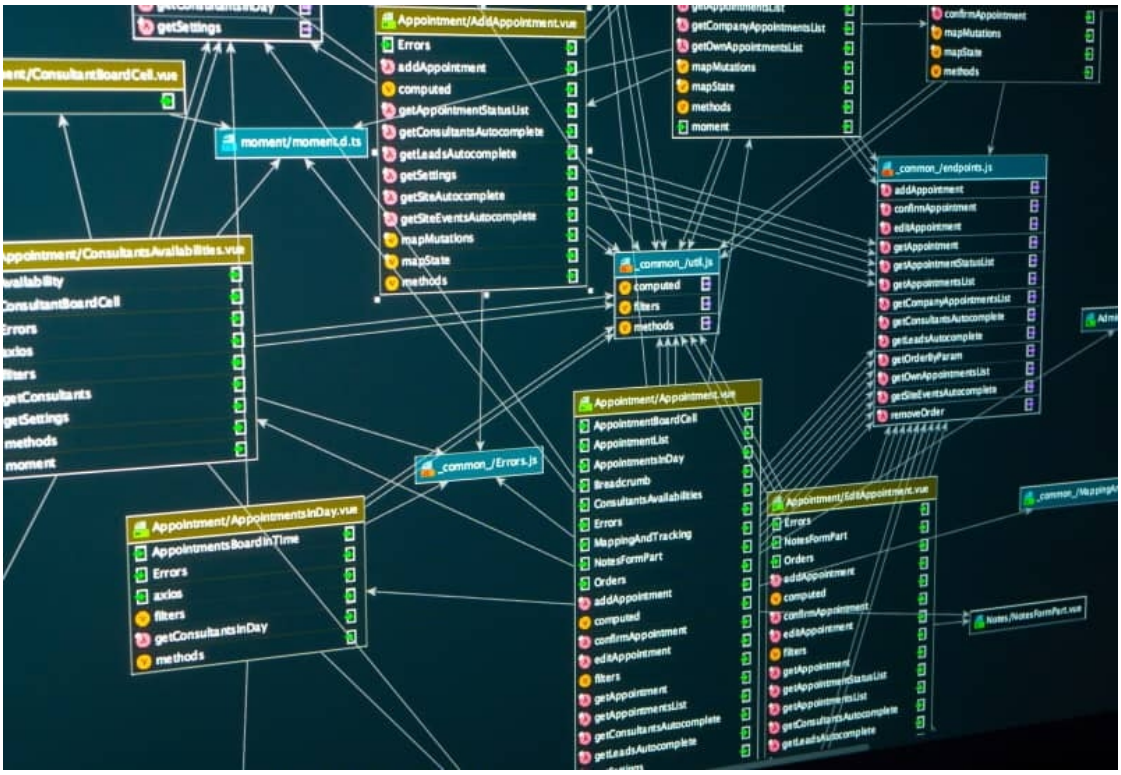
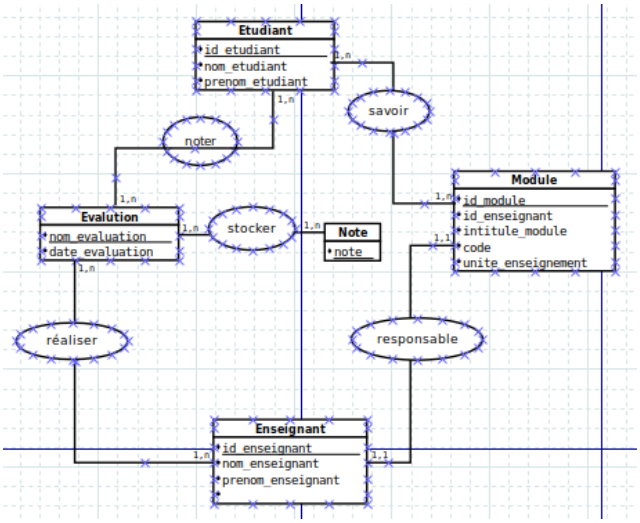
S1.04 : Création d'une base de données
2022-2023



SARAVANAN Bryan INDRA

2.1

1. Modèles entités-associations respectant la syntaxe du cours.



2. Schéma relationnel.

Étudiant (id_etudiant, nom_etudiant, prenom_etudiant)
Enseignant (id_enseignant, nom_enseignant, prenom_enseignant)
Module (id_module, intitule_module, code, unite_enseingnant, id_enseignant)
Évaluation (id_evaluation, nom_evaluation, date_evaluation)
Note (note, nom_evaluation, id_module, id_etudiant,)

3. Script SQL de création des tables.

```
CREATE TABLE Etudiant (
        id_etudiant INTEGER PRIMARY KEY,
        nom_etudiant VARCHAR (100),
        prenom_etudiant VARCHAR (100)
);

CREATE TABLE Enseignant (
         id_enseignant INTEGER PRIMARY KEY,
         nom_enseignant VARCHAR (100),
         prenom_enseignant VARCHAR (100)
);

CREATE TABLE Evaluation (
id_module INTEGER PRIMARY KEY,
        nom_evaluation VARCHAR (100) ,
        date_evaluation DATE
 foreign key (id_module) references module(id_module)
);

CREATE TABLE Module (
        id_module INTEGER PRIMARY KEY,
        intitule_module VARCHAR (100),
        code VARCHAR (100),
```

```
        unite_enseignant VARCHAR (100),
        id_enseignant integer,
        foreign key (id_enseignant) references Enseignant(id_enseignant)
);

CREATE TABLE Note (
        nom_evaluation VARCHAR (100) ,
        id_module INTEGER PRIMARY KEY ,
        id_enseignant integer not null,
        id_etudiant INTEGER NOT NULL,
        note FLOAT NOT NULL,
        foreign key (id_enseignant) references Enseignant(id_enseignant),
        foreign key (id_etudiant) references Etudiant(id_etudiant),
        foreign key (id_module) references module(id_module),


);
 create table copy(
        id_enseignant INTEGER,
         nom_enseignant VARCHAR (100),
         prenom_enseignant VARCHAR (100),
        id_module INTEGER,
        intitule_module VARCHAR (100),
        code VARCHAR (100),
        unite_enseignant VARCHAR (100),
        nom_evaluation VARCHAR (100),
        date_evaluation DATE,
        note FLOAT NOT NULL,
        id_etudiant INTEGER,
        nom_etudiant VARCHAR (100),
        prenom_etudiant VARCHAR (100));

// cette table me sert pour copier data.csv
```

2.2

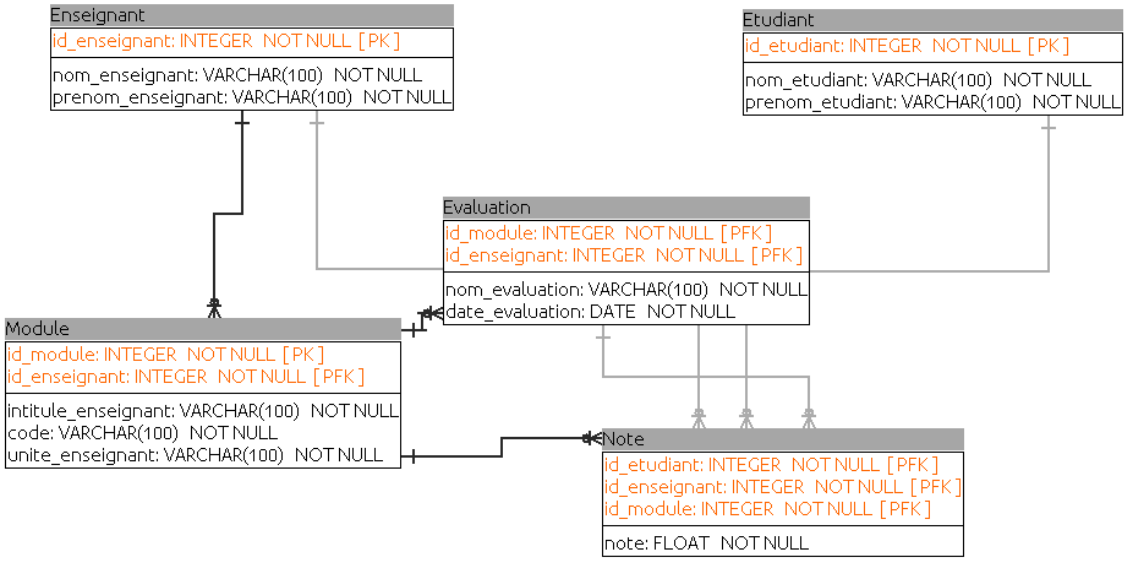
1. Illustrations comparatives cours/ AGL commentée d'une association fonctionnelle.

Association fonctionnelle, possède une clé étrangère lorsque il y a une cardinalité de 1,1.

2. Illustrations comparatives cours/ AGL commentée d'une association maillé.

Association maillée, jointure de deux tables différents avec un ou plusieurs clés étrangères, avec une cardinalité de 0, n. AGL, dans l'association maillée, il est présenté, mais pas représenter

3. Modèle entités-associations avec AGL.



4. Script SQL de création des tables.

```
CREATE SEQUENCE public.enseignant_id_enseignant_seq;

CREATE SEQUENCE public.enseignant_nom_enseignant_seq;

CREATE SEQUENCE public.enseignant_prenom_enseignant_seq;

CREATE TABLE public.Enseignant (
	id_enseignant INTEGER NOT NULL DEFAULT
nextval('public.enseignant_id_enseignant_seq'),
	nom_enseignant VARCHAR(100) NOT NULL DEFAULT
nextval('public.enseignant_nom_enseignant_seq'),
	prenom_enseignant VARCHAR(100) NOT NULL DEFAULT
nextval('public.enseignant_prenom_enseignant_seq'),
	CONSTRAINT id_enseignant PRIMARY KEY (id_enseignant)
);


ALTER SEQUENCE public.enseignant_id_enseignant_seq OWNED BY
public.Enseignant.id_enseignant;

ALTER SEQUENCE public.enseignant_nom_enseignant_seq OWNED BY
public.Enseignant.nom_enseignant;
```

```
ALTER SEQUENCE public.enseignant_prenom_enseignant_seq OWNED BY
public.Enseignant.prenom_enseignant;

CREATE SEQUENCE public.module_id_module_seq_1;

CREATE TABLE public.Module (
	id_module INTEGER NOT NULL DEFAULT
nextval('public.module_id_module_seq_1'),
	id_enseignant INTEGER NOT NULL,
	intitule_enseignant VARCHAR(100) NOT NULL,
	code VARCHAR(100) NOT NULL,
	unite_enseignant VARCHAR(100) NOT NULL,
	CONSTRAINT id_module PRIMARY KEY (id_module, id_enseignant)
);


ALTER SEQUENCE public.module_id_module_seq_1 OWNED BY public.Module.id_module;

CREATE SEQUENCE public.evaluation_date_evaluation_seq;

CREATE TABLE public.Evaluation (
	id_module INTEGER NOT NULL,
	id_enseignant INTEGER NOT NULL,
	nom_evaluation VARCHAR(100) NOT NULL,
	date_evaluation DATE NOT NULL DEFAULT
nextval('public.evaluation_date_evaluation_seq'),
	CONSTRAINT id_module PRIMARY KEY (id_module, id_enseignant)
);


ALTER SEQUENCE public.evaluation_date_evaluation_seq OWNED BY
public.Evaluation.date_evaluation;

CREATE SEQUENCE public.etudiant_id_etudiant_seq;

CREATE SEQUENCE public.etudiant_prenom_etudiant_seq;

CREATE TABLE public.Etudiant (
	id_etudiant INTEGER NOT NULL DEFAULT nextval('public.etudiant_id_etudiant_seq'),
	nom_etudiant VARCHAR(100) NOT NULL,
	prenom_etudiant VARCHAR(100) NOT NULL DEFAULT
nextval('public.etudiant_prenom_etudiant_seq'),
	CONSTRAINT id_etudiant PRIMARY KEY (id_etudiant)
);


ALTER SEQUENCE public.etudiant_id_etudiant_seq OWNED BY public.Etudiant.id_etudiant;

ALTER SEQUENCE public.etudiant_prenom_etudiant_seq OWNED BY
public.Etudiant.prenom_etudiant;

CREATE TABLE public.Note (
```

```
	id_etudiant INTEGER NOT NULL,
	id_enseignant INTEGER NOT NULL,
	id_module INTEGER NOT NULL,
	note REAL NOT NULL,
	CONSTRAINT nom_evaluation PRIMARY KEY (id_etudiant, id_enseignant,
id_module)
);


ALTER TABLE public.Note ADD CONSTRAINT enseignant_note_fk
FOREIGN KEY (id_enseignant)
REFERENCES public.Enseignant (id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Module ADD CONSTRAINT enseignant_module_fk
FOREIGN KEY (id_enseignant)
REFERENCES public.Enseignant (id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Note ADD CONSTRAINT module_note_fk
FOREIGN KEY (id_module, id_enseignant)
REFERENCES public.Module (id_module, id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Evaluation ADD CONSTRAINT module_evaluation_fk
FOREIGN KEY (id_module, id_enseignant)
REFERENCES public.Module (id_module, id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Note ADD CONSTRAINT evaluation_note_fk
FOREIGN KEY (id_module, id_enseignant)
REFERENCES public.Evaluation (id_module, id_enseignant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;

ALTER TABLE public.Note ADD CONSTRAINT etudiant_note_fk
FOREIGN KEY (id_etudiant)
REFERENCES public.Etudiant (id_etudiant)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

5. La différences entre les scripts produits manuellement et automatiquement.

Les différences entre le script fait manuellement et le script généré automatiquement par AGL, c'est que AGL modifie a la fin les contraintes des attributs avec ALTER TABLE, ALTER SEQUENCE, CREATE SEQUENCE.

2.3

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

/ script de copie dans la table de copy depuis le fichier data.csv

\copy copy(id_enseignant, nom_enseignant, prenom_enseignant, id_module, intitule_module,code,unite_enseignant,nom_evaluation,date_evaluation,note,id_etudiant,nom_etudiant,prenom_etudiant) from '/home/bryan/Bureau/data.csv' delimiter ';' CSV header;
COPY 6087

// script pour peuplement de la table etudiant depuis copy

insert into etudiant(id_etudiant,nom_etudiant,prenom_etudiant) select distinct id_etudiant,nom_etudiant,prenom_etudiant from copy;
INSERT 0 137

// script pour peuplement de la table enseignant depuis copy

mabase=> insert into enseignant(id_enseignant,nom_enseignant,prenom_enseignant) select distinct id_enseignant,nom_enseignant,prenom_enseignant from copy;

INSERT 0 18

// script pour peuplement de la table module depuis copy

insert into module(id_module,intitule_module,code,unite_enseignant,id_enseignant) select distinct id_module,intitule_module,code,unite_enseignant,id_enseignant from copy;

INSERT 0 18

// script pour peuplement de la table evaluation depuis copy

insert into evaluation(id_module,nom_evaluation, date_evaluation) select distinct id_module,nom_evaluation, date_evaluation from copy;

// script pour peuplement de la table note depuis copy

insert into note(nom_evaluation,id_module,id_enseignant,id_etudiant,note) select distinct nom_evaluation,id_module,id_enseignant,id_etudiant,note from copy;

2. Présentation commentée de deux rêquetes intéressantes sur la base de données.

Requête :

- select distinct * from etudiant order by id_etudiant asc limit 10;

```
 id_etudiant | nom_etudiant | prenom_etudiant
-------------+--------------+-----------------
           1 | Hanquez      | Rachid
           2 | Parchard     | Aziza
           3 | Faity        | Olivier
           4 | Remy         | Marie
           5 | Guillet      | Jerome
           6 | Franceschi   | Bruno
           7 | Schilling    | Marius
           8 | Nouqueret    | Benoit
          10 | Liagre       | Lucile
          11 | Levasseur    | Cyril

(10 rows)
```

- select * from module limit 10;

```
 id_module | intitule_module | code |                 unite_enseignant                  | id_enseignant
-----------+-----------------+------+---------------------------------------------------+---------------
        18 | R107            | UE12 | Outils mathématiques fondamentaux                 |           161
         2 | R101            | UE12 | Initiation au développement                       |           145
         3 | R102            | UE12 | Développement d'interfaces web                    |           146
        11 | S101            | UE13 | Implémentation d'un besoin client                 |           154
        17 | R109            | UE12 | Économie durable et numérique                     |           160
         9 | S106            | UE13 | Découverte de l'environnement économique et écologique |           152
         6 | R111            | UE12 | Bases de la communication                         |           149
        12 | R112            | UE12 | Projet professionnel et personnel                 |           155
        13 | S105            | UE13 | Recueil de besoins                                |           156
         8 | R108            | UE12 | Gestion de projet & des organisations             |           151
(10 rows)
```