Rapport SAE Base de données

ESSAFA Yassine
Notos BUT1 INFO

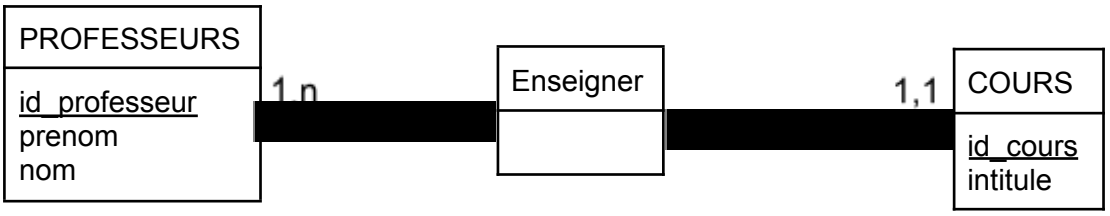
Sommaire :

Script SQL de création des tables :

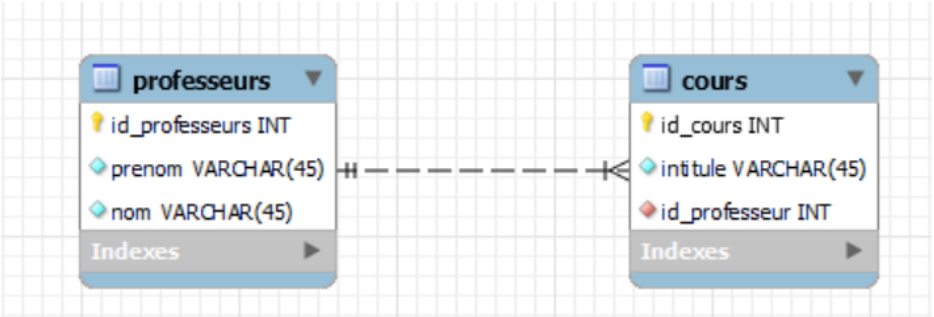
```
CREATE TABLE region (
  region_code INT PRIMARY KEY,
  region_name VARCHAR
);

CREATE TABLE status (
        status VARCHAR PRIMARY KEY
);

CREATE TABLE country (
        id_country INT PRIMARY KEY,
        country_name VARCHAR,
        region_code INT REFERENCES region(region_code),
        is_ldc INT
);

CREATE TABLE freedom (
        id_country SERIAL REFERENCES country(id_country),
        year INT,
        civil_liberties INT,
        political_rights INT,
        status VARCHAR REFERENCES status(status),
        PRIMARY KEY (id_country,year)
);
```

Illustrations association fonctionnelle :

MCD (modèle du cours) :



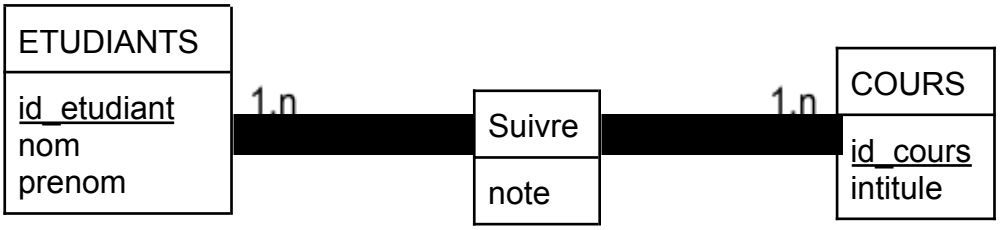
MPD (modèle de l'AGL) :



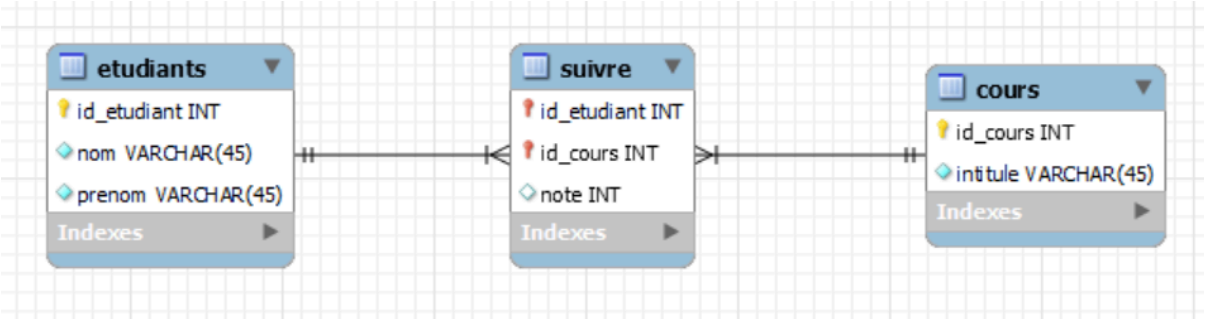
On remarque que l'entité Enseigner a été traduite par l'ajout d'une clé étrangère dans la table cours faisant référence à la clé primaire id_professeur de la table PROFESSEURS.

Illustrations association maillée :

MCD (modèle du cours) :

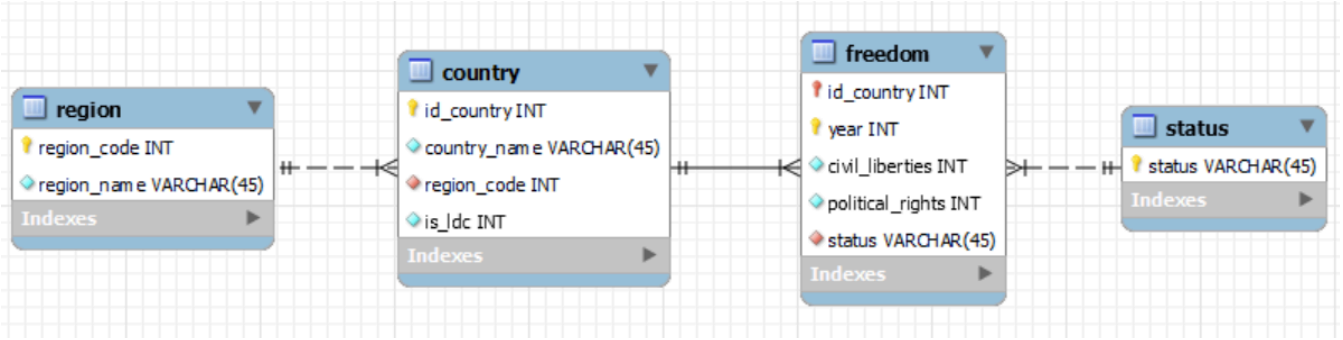


MPD (modèle de l'AGL) :



Cette fois, l'association est traduite par l'ajout de deux clé étrangères id_etudiant et id_cours dans la table Suivre faisant référence respectivement à la table ETUDIANTS et COURS. La combinaison de ces deux clés étrangères devient alors la clé primaire de la table Suivre.

MPD de la base de données de la SAE :



Script SQL généré par l'AGL :

```
CREATE TABLE `freedom`.`region` (
  `region_code` INT NOT NULL,
  `region_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`region_code`));

CREATE TABLE `freedom`.`status` (
  `status` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`status`));

CREATE TABLE `freedom`.`country` (
  `id_country` INT NOT NULL,
  `country_name` VARCHAR(45) NOT NULL,
  `region_code` INT NOT NULL,
  `is_ldc` INT NOT NULL,
  PRIMARY KEY (`id_country`),
  INDEX `region_code_idx` (`region_code` ASC) VISIBLE,
  CONSTRAINT `region_code`
        FOREIGN KEY (`region_code`)
```

```
        REFERENCES `freedom`.`region` (`region_code`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);

CREATE TABLE `freedom`.`freedom` (
  `id_country` INT NOT NULL,
  `year` INT NOT NULL,
  `civil_liberties` INT NOT NULL,
  `political_rights` INT NOT NULL,
  `status` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_country`, `year`),
  INDEX `status_idx` (`status` ASC) VISIBLE,
  CONSTRAINT `id_country`
        FOREIGN KEY (`id_country`)
        REFERENCES `freedom`.`country` (`id_country`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `status`
        FOREIGN KEY (`status`)
        REFERENCES `freedom`.`status` (`status`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);
```

Différences script manuel/automatique :

Le script généré automatiquement contient quelques différences avec celui produit manuellement. Tout d'abord, les clés primaires sont déclarées à part dans le script de l'AGL, même lorsqu'elles ne sont pas composées. De plus, l'AGL précise pour chaque attribut si la valeur NULL est acceptée ou non. La différence la plus importante est la façon dont sont implémentées les clés étrangères. En effet, au lieu de se contenter d'un simple REFERENCES, MySQL Workbench ajoute un index ainsi qu'une contrainte dans la création de la table. Enfin, l'AGL précise les actions à effectuer en cas de suppression ou de mise à jour d'une clé primaire référencée dans une autre table, ce qui est très important afin d'éviter des erreurs avec les clés étrangères qui pourraient pointer vers une valeur incorrecte.

Script de peuplement :

```
/* Création d'une table temporaire data qui accueillera les données du fichier csv */
CREATE TABLE data (
   country VARCHAR,
   year INT,
   cl INT,
```

```
    pr INT,
    status VARCHAR,
    region_code INT,
    region_name VARCHAR,
    is_ldc INT
);

/* Copie des informations contenues dans le fichier freedom.csv vers la table data
Pour que la requête suivante puisse fonctionner, la première ligne du fichier csv a dû
être supprimée car elle contenait le nom des colonnes */

\copy data FROM freedom.csv WITH CSV DELIMITER ',';

/* Projection des données de data dans les différentes tables créées précédemment
Pour la table country, une modification a été effectuée pour la création de la table : le
type de id_country a été changé de INT à SERIAL */
INSERT INTO region SELECT DISTINCT region_code,region_name FROM data;
INSERT INTO status SELECT DISTINCT status FROM data;
INSERT INTO country (country_name,region_code,is_ldc) SELECT DISTINCT
country,region_code,is_ldc FROM data;
INSERT INTO freedom SELECT DISTINCT id_country,year,cl,pr,status FROM data
JOIN country ON data.country = country.country_name;

/* Suppression de la table data, car elle ne fait pas partie de la bases de données */
DROP TABLE data;
```