Safiya Nguyen BUT 1 Boreas

Rapport SAE 201
Création d’une base de données
Freedom in the world

2024

Sommaire

I. Script manuel de création de la base de données.....3
II. Modélisation et script de création avec MySQL Workbench..... 4
III. Peuplement des tables.....10

Dans le cadre de la SAE 201, j'ai été amené à utiliser le SGBD postgresql ainsi que l'AGL MySQL Workbench.
Dans un premier temps, j'ai réalisé un script manuel de création de table. Ensuite, j'ai modélisé et généré des scripts grâce à l'AGL. Enfin, j'ai inséré les données dans les tables.

I. Script manuel de création de la base de données

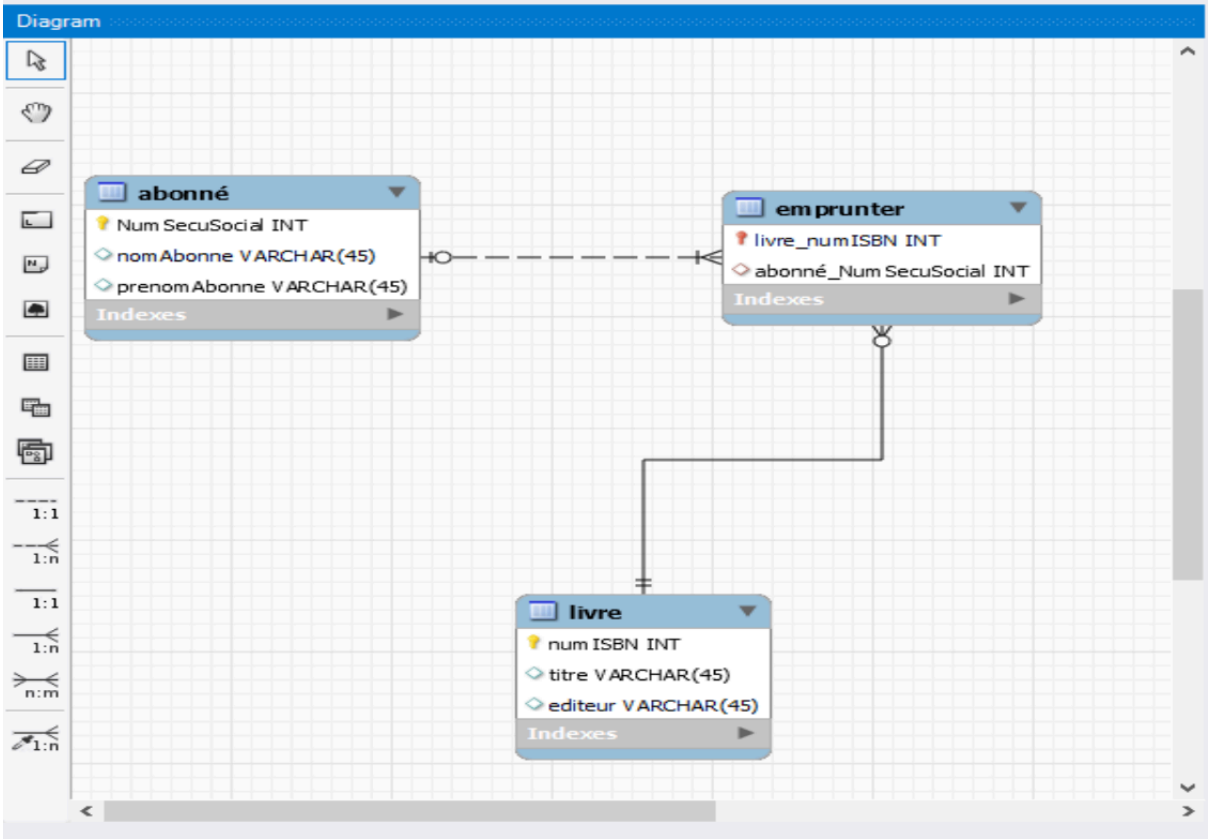
Voici le script de création de la base de données freedom in the world.

```
DROP TABLE freedom IF EXIST;
DROP TABLE country IF EXIST;
DROP TABLE status IF EXIST;
DROP TABLE region IF EXIST;

CREATE TABLE region(
     region_code INT PRIMARY KEY NOT NULL,
     name CHAR);

CREATE TABLE status(
     status VARCHAR(2)PRIMARY KEY NOT NULL);

CREATE TABLE country(
     id_country INT PRIMARY KEY NOT NULL,
     name CHAR,
     region_code  INT  REFERENCES  region(region_code)  ON
DELETE CASCADE,
     is_ldc INT);

CREATE TABLE freedom(
     id_country  INT  REFERENCES  country(id_country)  ON
DELETE CASCADE NOT NULL,
     year INT,
     civil_liberties  INT  CHECK(civil_liberties  >=  0  and
civil_liberties <= 7),
     political_rights INT CHECK(political_rights >= 0 and
political_rights <= 7),
     status VARCHAR(2) REFERENCES status(status) ON DELETE
CASCADE,
     PRIMARY KEY(id_country, year);
```

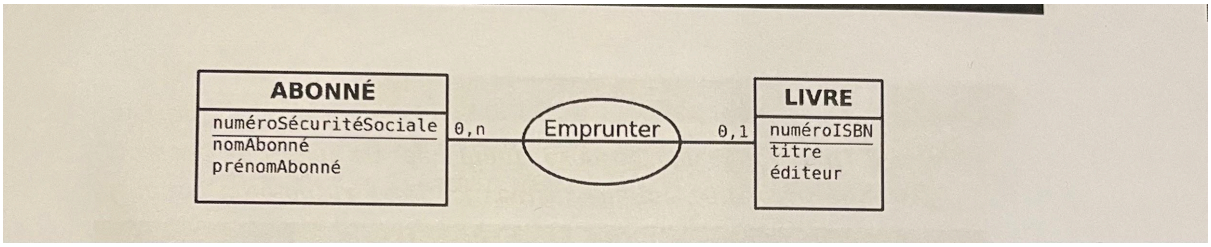
II. Modélisation et script de création avec MySQL Workbench

Pour m'aider à me familiariser avec l'AGL, j'ai modélisé une association fonctionnelle et une association maillée avec MySQL Workbench.

Ci-dessus vous trouverez un exemple d'association fonctionnelle réalisée grâce à l'AGL.



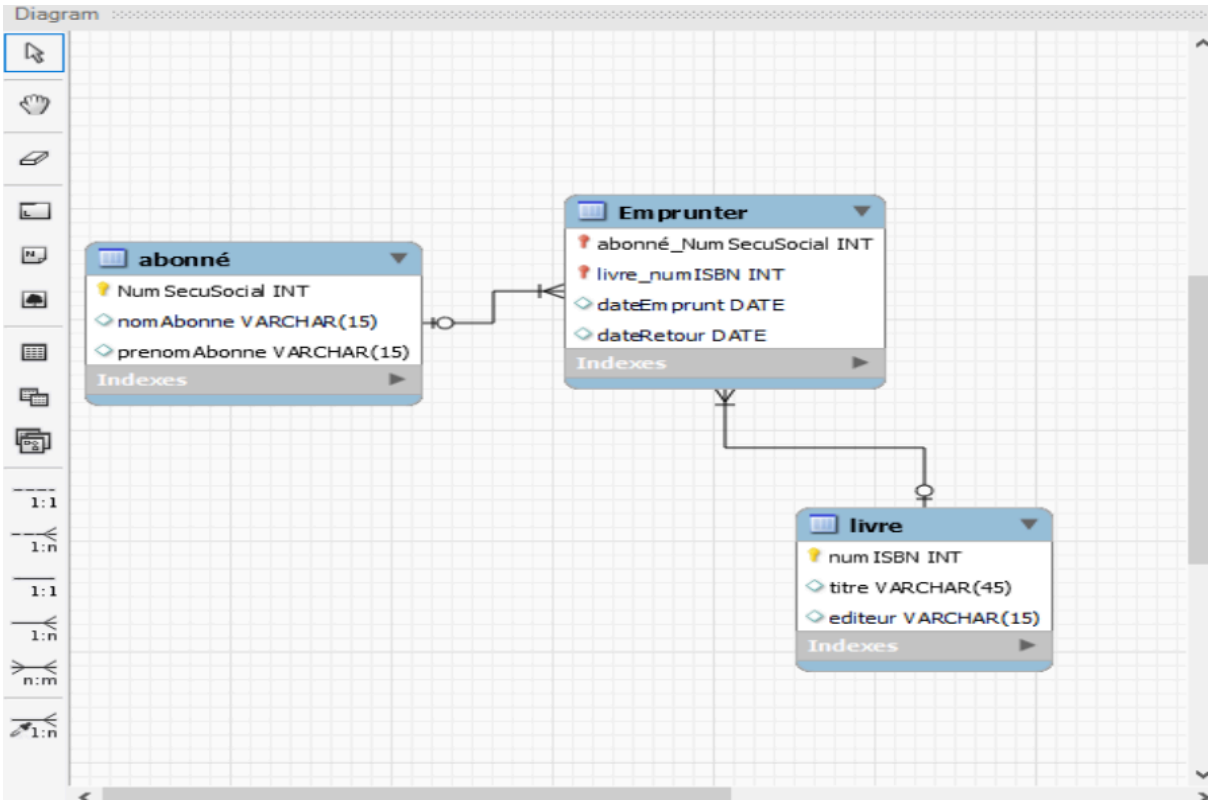
Ci-dessous un exemple de modèle fonctionnelle repris du cours.



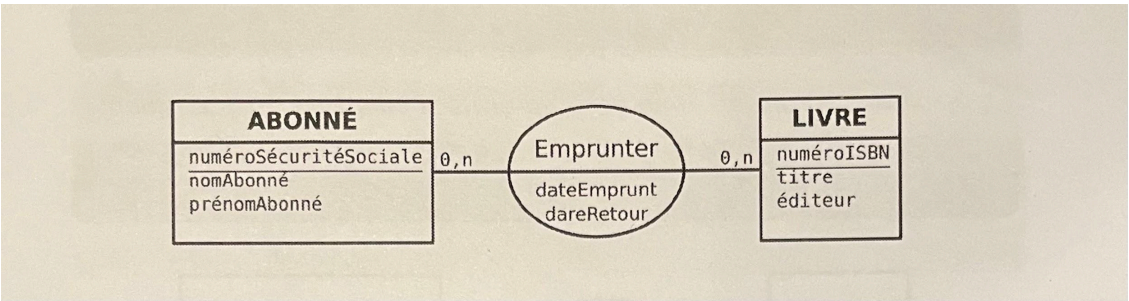
Similarité	Différence
- Les modèles sont ressemblants, on voit des types entité/association représentés en bloc	- On remarque qu'il y a des petites balises pour signaler les clés primaires et un losange rouge pour la clé étrangère.

- On retrouve les mêmes cardinalités	- Les cardinalités ne sont pas précisées en chiffres mais avec des signes. - Les types associations ont la même apparence que les types entités - On précise la nature des colonnes - Les identifiants ne sont pas implicites

Ci dessous vous trouverez un exemple d'association maillée réalisée avec l'AGL.

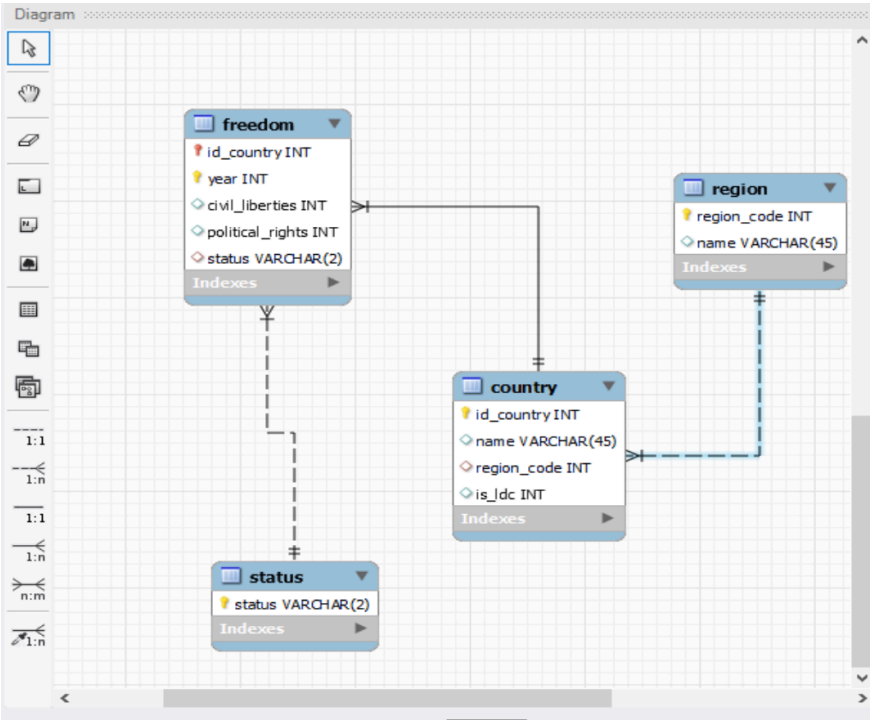


Ci-dessous vous trouverez un exemple d'association maillée repris du cours.



Similarité	Différence
- Le modèle de l'AGL reprend celle du cours	- Ici aussi, les identifiants sont explicites. - Les cardinalités ne sont pas symbolisés par des chiffres - Les types associations sont de la même forme que les types entités - Les identifiants ne sont pas soulignés mais symbolisés

Voici un modèle physique de données correspondant à la figure 2 de l'énoncé.



Après avoir modélisé la base de données, j'ai pu générer un script via l'AGL.

```
-- MySQL Script generated by MySQL Workbench
-- Sat Jan 20 14:56:26 2024
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET           @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET                                 @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_I
N_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema sae
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Table `region`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `region` (
  `region_code` INT NOT NULL,
  `name` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`region_code`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `country`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `country` (
  `id_country` INT NOT NULL,
  `name` VARCHAR(45) NULL DEFAULT NULL,
  `region_code` INT NULL DEFAULT NULL,
```

```
  `is_ldc` INT NULL DEFAULT NULL,
  PRIMARY KEY (`id_country`),
  INDEX `region_code_idx` (`region_code` ASC) VISIBLE,
  CONSTRAINT `region_code`
    FOREIGN KEY (`region_code`)
    REFERENCES `region` (`region_code`)
    ON DELETE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `status`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `status` (
  `status` VARCHAR(2) NOT NULL,
  PRIMARY KEY (`status`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `freedom`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `freedom` (
  `id_country` INT NOT NULL,
  `year` INT NOT NULL,
  `civil_liberties` INT NULL DEFAULT NULL,
  `political_rights` INT NULL DEFAULT NULL,
  `status` VARCHAR(2) NULL DEFAULT NULL,
  PRIMARY KEY (`id_country`, `year`),
  INDEX `status_idx` (`status` ASC) VISIBLE,
  CONSTRAINT `id_country`
    FOREIGN KEY (`id_country`)
    REFERENCES `country` (`id_country`)
    ON DELETE CASCADE,
  CONSTRAINT `status`
    FOREIGN KEY (`status`)
    REFERENCES `status` (`status`)
```

```
    ON DELETE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

On remarque que le script généré par l'AGL est beaucoup plus long que celui que j'ai écrit. Au début, on voit qu'il y a quelques paramètres de configuration qui sont désactivés et qui seront réactivés à la fin du script. Au début de mon script, j'ai supprimé chaque table une à une au cas où elles existaient déjà alors que l'AGL le fait au fur et à mesure : `CREATE TABLE IF NOT EXISTS `freedom`.` J'ai précisé directement quelles étaient les clés primaires en mettant les attributs et la précision alors que le script le mentionne à la fin de la création de table, de même pour les clés étrangères. On remarque qu'après avoir créé une table, il est spécifié le moteur de stockage, le jeu de caractère par défaut et la collation (comment les caractères sont triés et stockés). Il est précisé qu'il y a des valeurs par défaut (`NULL`).
Lors de la création des tables country et freedom, un index a été créé afin d'accélérer les opérations de recherches. On peut également voir que la contrainte des clés étrangères a été appliquée.

III. Peuplement des tables

Tout d'abord, je crée une table temporaire avec les données du fichier plat:

- clic droit sur le schéma dans lequel on veut la table et cliquer sur Table data import wizard puis on crée une nouvelle table que je nomme temporaire. Ensuite, je suis les instructions.
- Je remplis les tables à partir de la table temporaire:

```
INSERT INTO sae.region
SELECT DISTINCT Region_Code, Region_Name
FROM sae.temporaire;

INSERT INTO sae.status
```

```
SELECT DISTINCT status
FROM sae.temporaire;
```

- Pour la table country, étant donné qu'on n'a pas son identifiant, on va changer le type de son identifiant qui était `INT` à la base en `SERIAL`. Ce qui va permettre d'incrémenter automatiquement les identifiants.

```
INSERT INTO sae.country
SELECT DISTINCT country, Region_Code, is_ldc
FROM sae.temporaire;
```

J'ai essayé d'insérer de cette manière cependant, ça ne fonctionnait pas. Ne trouvant pas l'erreur, je vous montre la façon dont j'aurais inséré les valeurs dans freedom:

```
INSERT INTO sae.freedom
SELECT DISTINCT country.id_country, temporaire.year,
temporaire.CL, temporaire.PR, temporaire.status
FROM sae.temporaire
JOIN country ON country.name = temporaire.country;
```