

SAE Création d'une base de données : Freedom in the world**Sommaire:****1 Contexte de la SAE****2 Travail demandé****2.1 Script manuel de création de la base de données****2.2 Modélisation et script de création "avec AGL"****2.3 Peuplement des tables****1 Contexte de la SAE**

L'objectif de cette SAE est de concevoir, mettre en place puis alimenter une base de données permettant de stocker des mesures de sentiment de degré de liberté et de droits politiques par pays.

2 Travail demandé**2.1 Script manuel de création de la base de données**

Pour commencer, on crée les tables qui correspondent à la modélisation entités-associations de la base de donnée:

```
CREATE TABLE region (  
    region_code INT PRIMARY KEY,  
    name VARCHAR(255)  
);
```

Cette requête crée une table "region" avec deux colonnes : "region_code" qui est une clé primaire de type entier, et "name" qui est une chaîne de caractères pour stocker le nom de la région.

```
CREATE TABLE country (  
    id_country INT PRIMARY KEY,  
    name VARCHAR(255),  
    is_ldc BOOLEAN,  
    region_code INT REFERENCES region);  
);
```

Cette requête crée une table "country" avec quatre colonnes : "id_country" qui est une clé primaire de type entier, name qui est une chaîne de caractères pour stocker le nom du pays, is_ldc qui est un booléen qui permet de savoir si un pays est développé ou non, et une clé étrangère "region_code" faisant référence à la table "region".

```
CREATE TABLE freedom (  

```

```

id_country INT,
year INT,
civil_liberties INT,
political_rights INT,
status VARCHAR(255),
PRIMARY KEY (id_country, year),
FOREIGN KEY (id_country) REFERENCES country(id_country),
FOREIGN KEY (status) REFERENCES status(status)
);

```

Cette requête crée une table “freedom” avec year comme clé primaire de type INT et deux clés étrangères qui sont status et id_country. Ces clés étrangères établissent des relations avec les tables "country" et "status"

```

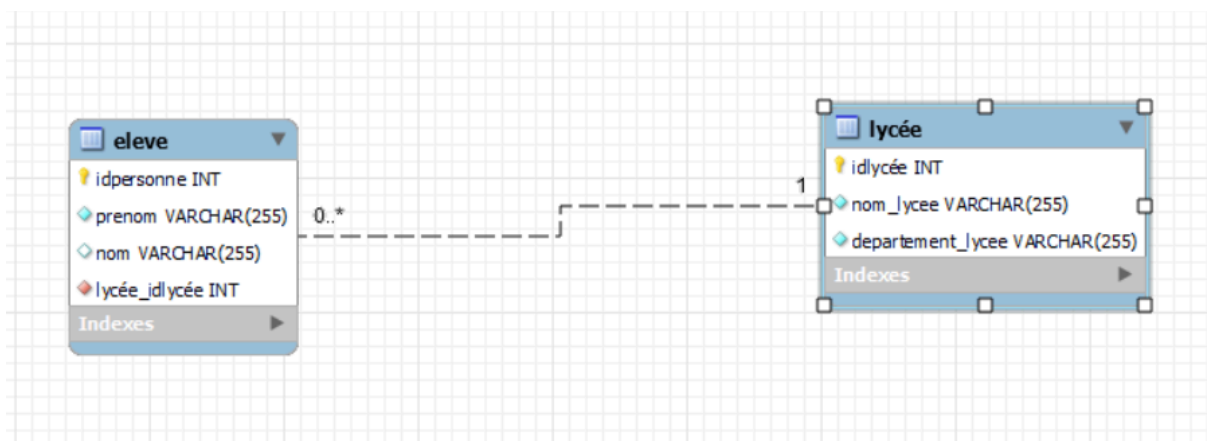
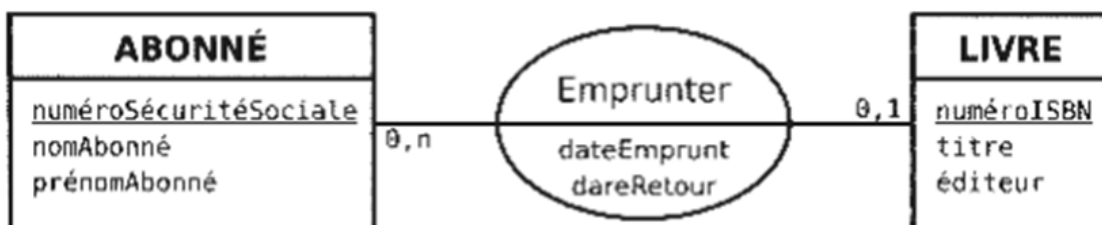
CREATE TABLE status (
    status VARCHAR(255) PRIMARY KEY
);

```

Cette requête crée une table “status” avec une colonne status qui est une clé primaire de type chaîne de caractère qui peut être soit NF, PF ou F qui correspond à leur niveau de liberté.

2.2 Modélisation et script de création “avec AGL”

1) Illustrations comparatives cours/AGL commentée d’une association fonctionnelle



Avec l'AGL, vous pouvez examiner chaque attribut en détaillant son nom, son type (INT,

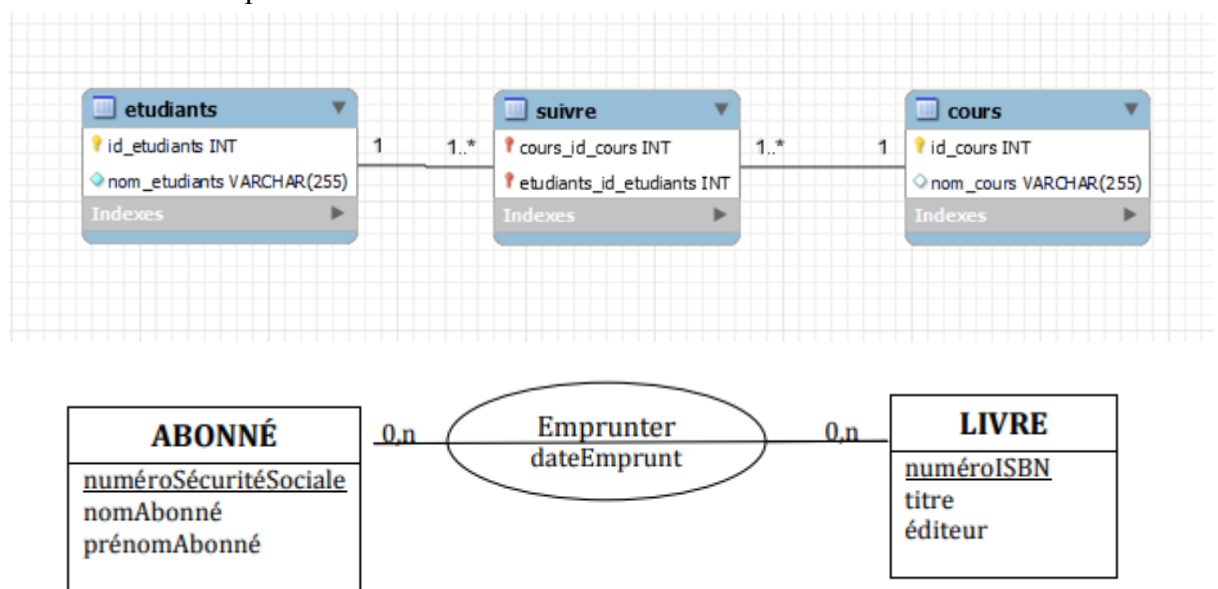
VARCHAR, etc.) et utiliser des symboles distincts. Par exemple, une clé primaire est représentée par une clé jaune (NUMSECU), tandis qu'une clé étrangère est indiquée par un point rouge.

En Modèle Conceptuel de Données (MCD), seuls les libellés des attributs sont visibles, et les clés étrangères ne sont pas explicitement illustrées. La clé primaire est généralement mise en évidence par un soulignement. Les associations et les cardinalités sont définies, par exemple, "0,1" signifie qu'une entité peut avoir zéro ou au maximum une relation, tandis que "0, n" signifie zéro ou plusieurs relations.

En ce qui concerne les associations et les cardinalités dans l'AGL, les emplacements des cardinalités sont inversés par rapport au MCD. Cependant, le concept de nombre possible de relations entre les classes reste le même.

Les deux entités sont reliées par une ligne droite et un type d'association, avec des cardinalités telles que '0,n' et '0,1'. La cardinalité "0,1" indique qu'une entité peut ne pas avoir de relation du tout avec une autre entité, ou au maximum, elle peut avoir une seule relation. La cardinalité "0,n" indique qu'une entité peut ne pas avoir de relation du tout avec une autre entité, ou elle peut avoir plusieurs relations.

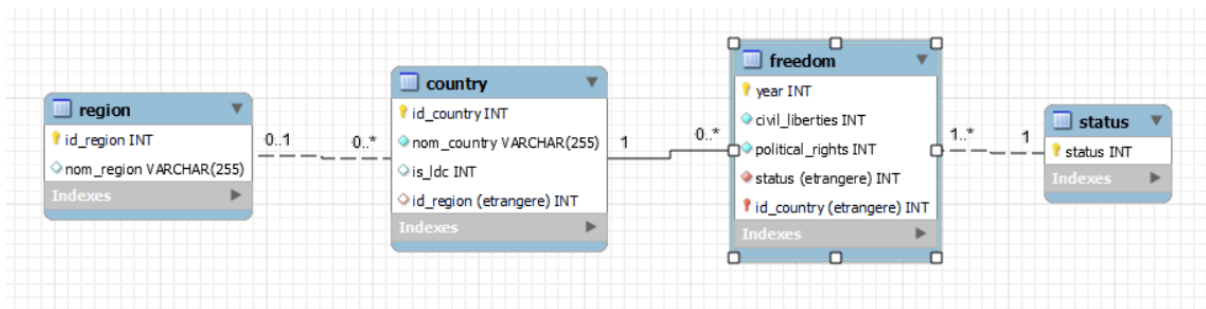
2) Illustrations comparatives cours/AGL commentée d'une association maillée



En ce qui concerne les attributs, les mêmes différences que celles observées pour l'association fonctionnelle sont notables. Pour ce qui est des associations et des cardinalités, les "0,n" sont affichés des deux côtés de la relation, ce qui indique que les deux entités peuvent avoir aucune relation ou une infinité de relations entre elles.

Le schéma de l'AGL présente quatre cardinalités, tandis que le MCD en comporte deux. Dans l'AGL, pour l'association maillée, une nouvelle classe est automatiquement créée, et les clés primaires des deux entités deviennent à la fois des clés primaires et étrangères de cette nouvelle classe.

3) Modèle physique de données réalisé avec l'AGL



On peut remarquer certaines ressemblances entre le modèle MCD et l'AGL. Les cardinalités 1 correspondent aux 1.1 ; 1.n = 1.* ; 0.n = 0.* ; 0.1 = 0.1.

4) Script SQL de création de table généré automatiquement par l'AGL

```
CREATE TABLE IF NOT EXISTS `mydb`.`freedom` (
  `year` INT NOT NULL,
  `civil_liberties` INT NOT NULL,
  `political_rights` INT NOT NULL,
  `status (etrangere)` INT NOT NULL,
  `id_country (etrangere)` INT NOT NULL,
  PRIMARY KEY (`year`, `id_country (etrangere)`),
  INDEX `fk_freedom_status1_idx` (`status (etrangere)` ASC) VISIBLE,
  INDEX `fk_freedom_country1_idx` (`id_country (etrangere)` ASC) VISIBLE,
  CONSTRAINT `fk_freedom_status1`
    FOREIGN KEY (`status (etrangere)`)
    REFERENCES `mydb`.`status` (`status`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_freedom_country1`
    FOREIGN KEY (`id_country (etrangere)`)
    REFERENCES `mydb`.`country` (`id_country`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

5) Discussion sur les différences entre les scripts produits manuellement et automatiquement

La création de scripts SQL, que ce soit manuellement sur un shell (en utilisant une interface en ligne de commande ou un outil graphique) ou automatiquement à partir d'un Atelier de Génie Logiciel (AGL), peut présenter quelques différences selon le contexte. Voici quelques points de comparaison:

- Manuel (Shell) : Lorsque vous écrivez des scripts SQL manuellement, vous saisissez chaque instruction SQL à la main en fonction des besoins du système.

- Automatique (AGL) : Les outils d'AGL peuvent générer automatiquement des scripts SQL en se basant sur des modèles de données ou des modèles conceptuels créés visuellement dans l'AGL. Cela peut être plus rapide et moins sujet aux erreurs humaines.
- Manuel (Shell) : Il n'y a généralement pas de représentation visuelle associée. Les instructions SQL sont écrites directement en texte.
- Automatique (AGL) : Vous pouvez utiliser des diagrammes de modèles de données dans l'AGL pour définir des entités, des relations et d'autres éléments. L'AGL peut ensuite générer automatiquement les scripts SQL correspondants.
- Manuel (Shell) : Il peut être plus facile de contrôler chaque détail, mais cela peut également conduire à des erreurs si la complexité du schéma de la base de données augmente.
- Automatique (AGL) : Les outils d'AGL visent à maintenir la cohérence entre le modèle visuel et les scripts générés, mais la complexité de la génération automatique peut entraîner des ajustements manuels.
- Manuel (Shell) : Les développeurs peuvent optimiser les requêtes manuellement en fonction des besoins spécifiques.
- Automatique (AGL) : La génération automatique peut produire des scripts standard qui peuvent nécessiter une optimisation manuelle pour des performances maximales.

2.3 Peuplement des tables

Tout d'abord, on lance PostgreSQL. Pour peupler la base de données à partir du fichier csv, on va créer la table temp contenant toutes les données

```
CREATE TABLE temp (
    country_name VARCHAR,
    year INT,
    civil_liberties INT,
    political_rights INT,
    status VARCHAR,
    region_code INTEGER,
    region_name VARCHAR,
    is_idc BOOLEAN
);
```

Liste des relations			
SchÚma	Nom	Type	PropriÚtaire
public	country	table	postgres
public	freedom	table	postgres
public	region	table	postgres
public	status	table	postgres
public	temp	table	postgres

(5 lignes)

Ensuite, on copie les données du fichier dedans avec:

```
\copy temp FROM 'C:\Users\Moilim\OneDrive\Bureau\freedom.csv' DELIMITER ','
CSV HEADER;
```

Cela nous renvoie "COPY 4979", c'est-à-dire que le transfert s'est bien déroulé.

Ensuite on insère les données dans les différentes tables:

Insertion des Données dans la Table Region :

```
INSERT INTO region (region_code, name)
```

```
SELECT DISTINCT region_code, region_name FROM temp;
```

Les données uniques des colonnes "region_code" et "region_name" de la table temporaire "temp" sont insérées dans la table "region".

```
postgres=# SELECT * FROM region;
 region_code | name
-----+-----
          142 | Asia
           9 | Oceania
          19 | Americas
         150 | Europe
           2 | Africa
(5 lignes)

postgres=#
```

Insertion des Données dans la Table Country :

```
INSERT INTO country (name, region_code, is_ldc)
```

```
SELECT DISTINCT country_name, region_code, is_ldc FROM temp;
```

Les données uniques des colonnes "country_name", "region_code", et "is_ldc" de la table temporaire "temp" sont insérées dans la table "country".

```
postgres=# SELECT * FROM country;
 id_country | country | region_
code | is_ldc
-----+-----+-----
150 | f | 1 | Croatia |
9 | f | 2 | Nauru |
150 | f | 3 | Montenegro |
142 | t | 4 | Yemen |
150 | f | 5 | Romania |
150 | f | 6 | Philippines |
```

Insertion des Données dans la Table Status :

```
INSERT INTO status (status)
```

```
SELECT DISTINCT Status FROM temp;
```

Les données uniques de la colonne "Status" de la table temporaire "temp" sont insérées dans la table "status".

```
postgres=# SELECT * FROM status;
status
-----
PF
NF
F
(3 lignes)
```

Insertion des Données dans la Table Freedom :

```
INSERT INTO freedom (year, id_country, civil_liberties, political_rights, status)
```

```
SELECT year, country.id_country, civil_liberties, political_rights, status
```

```
FROM temp
```

```
JOIN country ON temp.country_name = country.name;
```

Cette étape réalise une jointure entre les tables "temp" et "country" sur la base du nom du pays, puis insère les données correspondantes dans la table "freedom".

```
postgres=# SELECT * FROM freedom;
```

id_country	year	civil_liberties	political_rights	status
16	1995	7	7	NF
16	1996	7	7	NF
16	1997	7	7	NF
16	1998	7	7	NF
16	1999	7	7	NF
16	2000	7	7	NF
16	2001	7	7	NF
16	2002	6	6	NF
16	2003	6	6	NF
16	2004	6	5	NF
16	2005	5	5	PF
16	2006	5	5	PF
16	2007	5	5	PF